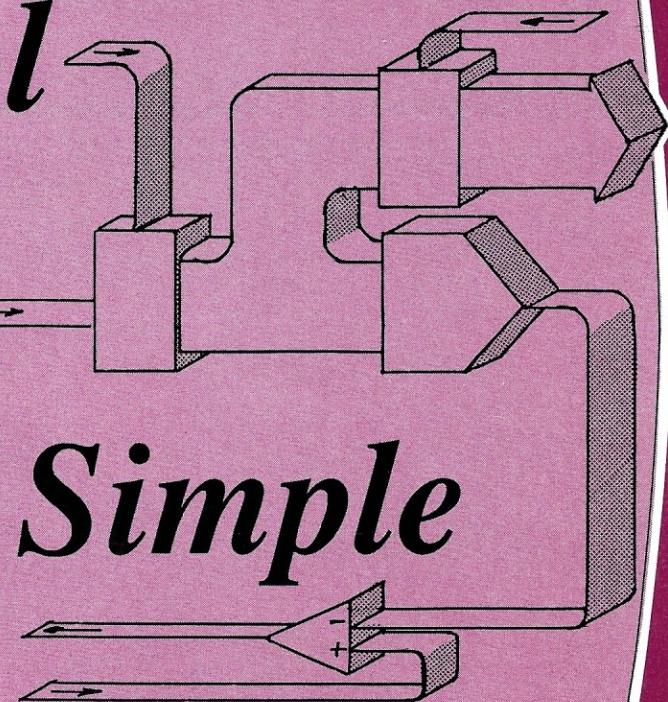


For the Advanced Computer User

Micro/Systems Journal™

*Control
Systems
Made Simple*

see pages 20 - 24



Also in this Issue

A First Look At The 80386	34
Interfacing Using The SCSI Bus	46
Program Interfacing To MS-DOS	62
Using SYMDEB With NMI Breakpoints	66
SYSLIB, Z3LIB & VLIB	72

Complete Table of Contents on Page 3

September/October 1986

Vol.2/No.5

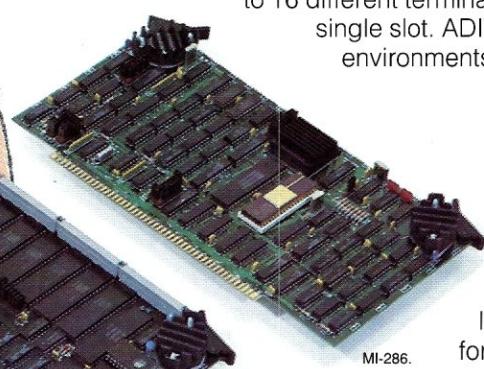
\$4.50
U.S.A.

MACROTECH—STILL THE S-100 PERFORMANCE PACESETTER



MI-286. Our 80286/Z80H Dual CPU Board is at least twice as fast as Compupro's 8085/88 and it's a direct replacement. The MI-286 has already become the standard by which other 80286 based systems are measured. Ask us for a complimentary Benchmark Report.

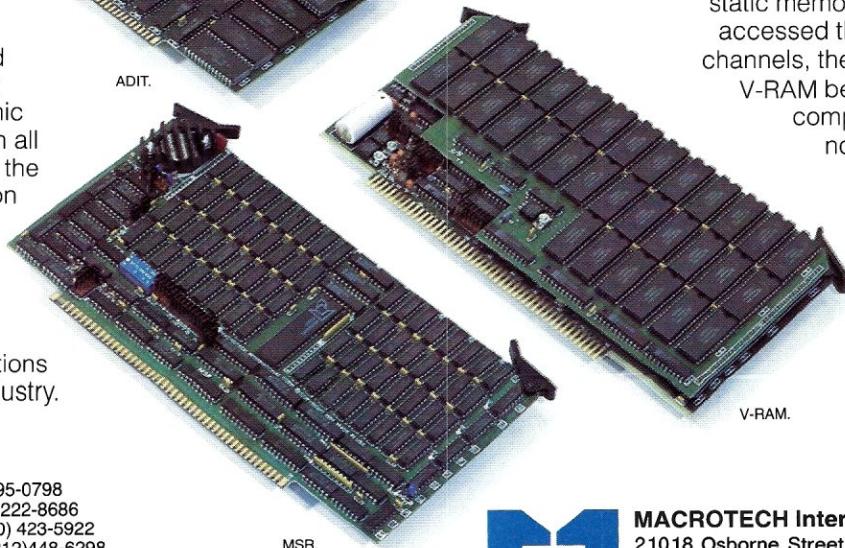
ADIT. There's nothing else like it on the market. It's an Intelligent I/O Board with its own real time firmware that lets you control up to 16 different terminals, modems or printers all from a single slot. ADIT is the performance standard in environments such as Alpha Micro where I/O speed is critical.



MI-286.

ADIT.

MSR. High performance and reliability in a memory so fast you won't believe it's a dynamic ram product. Compatible with all popular S-100 environments, the MSR's low power consumption and 120 nanosecond ram devices set a new standard for dynamic memory products. The MSR is available in quarter, half, one and two megabyte configurations at the lowest prices in the industry.



MSR.

V-RAM.



MACROTECH International Corp.
21018 Osborne Street
Canoga Park, CA 91304
(800) 824-3181 • in Calif. (818) 700-1501
Telex: 9109970653

Dealers:

Gifford Computer Systems (415) 895-0798
Custom Computer Technology (800) 222-8686
S-100 (800) 423-5922
John D. Owens & Associates (212) 448-6298
In England; Fulcrum (Europe) Ltd. (0621) 828763

Macrotech dealers also include most Compupro Systems Centers, Heathkit Electronic Centers and Alpha Micro Dealers.

MORE PERFORMANCE

In a multi-user system, performance is the most important ingredient.

Introducing the top-of-the-line in our 286 series, the **CompuPro 286/80™** with more performance built in than ever. For starters, our new 8 MHz 80286 processor gets data from memory in only two clock cycles, compared with the three or four required by other systems. In addition to the 768K of high speed memory, the system's high performance 80 Megabyte hard disk has a dedicated 512K byte cache buffer that boosts its performance even more. Tape back-up and floppy drives are included along with nine serial ports, Concurrent DOS 816™ multi-user multi-tasking operating system and NewWord™ word processing software. The system includes a 16 slot S-100R motherboard, which means there's plenty of room for adding options such as memory expansion to a megabyte, more cache memory, additional I/O ports, slave processors, graphics, networking and more.

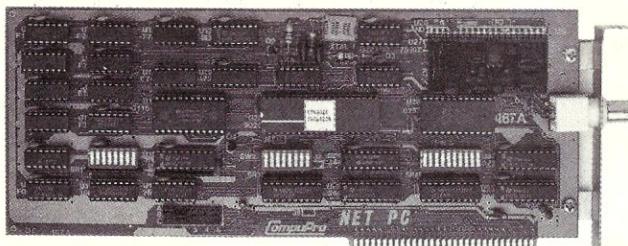
All this tech-talk translates simply to this: The CompuPro 286/80 has the power to solve your business computing needs today and tomorrow as your business grows.

The price? A lot lower than you'd expect to pay for all this performance. Just \$12,500 suggested list. To find out how the CompuPro 286/80 (or any of our other multi-user systems) can give your business more performance, call today for the name of your nearest CompuPro dealer.

10 MHz available with **AMD 80286!** Dealers have details.



MORE NET PERFORMANCE



CompuPro has been providing networking solutions for years with our **NET 100™** and **NET 10™** products, and we're proud to introduce the latest member of our network family—The **NET PC™**. Simply plug the NET PC into your IBM® PC, XT, AT or 100% compatible and your personal computer now becomes an integrated part of a high performance multi-user network. The **NET PC** includes the CPLINK™ requestor program that lets your PC become a part of the network while still running DOS.

CompuPro multi-user systems (like the 286/80) make ideal file-servers for networked PCs, and they're still fully functional as to a true multi-user system and keep your existing PCs is now solved.

multi-user computers. Your problem of how to move up

TECH-TALK: All our network products use the high speed (2.5 MBPS) ARCNET® token passing hardware protocol and DR NET™ software protocols (except a PC with CPLINK, which emulates DOS 3.1 conventions). The network may contain up to 255 nodes. The NET PC is a fully compatible super-set of the SMC ARCNET card and therefore may be used with other network software systems.

NET PC with CPLINK: \$550. NET 100 (for S-100 systems and the 286 series): \$495. NET 10 (for the CompuPro 10 Plus™): \$495.
All prices are suggested list.

AT COMPUPRO, PERFORMANCE IS NOT JUST ANOTHER BUZZWORD.

SUMMER HOT FLASH! Get a super summer deal on a CompuPro 10 Plus four user system with a 40 megabyte hard disk. For a limited time we're slashing the price to just \$5995! See your nearest dealer for more details.

CompuPro™

Viasyn Corporation, 26538 Danti Court
Hayward, CA 94545-3999 (415) 786-0909

CompuPro, 286/80, 10 Plus, NET 100, NET 10 and NET PC are trademarks of Viasyn Corporation. Concurrent DOS 816 is compound trademark of Digital Research, Inc. and Viasyn Corporation. DR NET is a trademark of Digital Research, Inc. NewWord is trademark of NewStar Software, Inc. CPLINK is a trademark of ComputerNet A/S. IBM is registered trademark of International Business Machines Corporation. ARCNET is a registered trademark of Datapoint Corporation. Prices and specifications subject to change without notice.

MIX C COMPILER

Works on all
MSDOS/PCDOS
and all CP/M Z80
COMPUTERS!
(Not Copy
Protected!)

The Powerful Mix
C COMPILER
Harness the Power of the C Language
with this full featured Compiler
MIX Software Inc.
Software for the Future

Incredible Value
\$39.95
AT ONLY With 30 Day
Money-Back Guarantee

Split Screen Text Editor

an Incredible
Value **\$29.95**
AT ONLY

Our high powered editor is great for editing high level languages. It works just like Micropro's WordstarTM but macros allow you to create your own custom editor, and the split-screen feature lets you edit two files at the same time.



The MSDOS/PCDOS version is loaded with special features:

- Execute any DOS command or RUN other programs from the editor.
- Quickly edit files as large as 300,000 characters.
- Compile MIX C programs directly from memory. The editor automatically positions the cursor to the first error in your program.

ASM UTILITY
an Incredible **\$10**



Call assembly language routines from your C programs. The ASM utility works with Microsoft's MASM or M80 assembler. Macros make it easy! Works just as if you were calling a C function, and you can even call C functions from assembly language. Lots of useful assembly language functions are included as examples.

C FEVER catch it!

It's becoming an epidemic . . . everyone is switching to C! First there were a few hackers, then came the college students, next the major software houses, and now the rest of the programming world. Programmers everywhere are infected with the desire for SPEED, POWER, and PORTABILITY.

It's time to face the inevitable. You're going to catch the fever too! When you do, give us a call. We've got the best cure—an illustrated guide to the C language, plus a complete program development system. Everything you need to master the C programming language . . . all at a price that's less than the cost of a book!

But don't let this price fool you. Our system is powerful; it compiles twice as fast as the others, is completely standard, and it's very easy to use. Most C compilers are designed for wizards. We have designed ours for you!

What do you get for a mere \$39.95?

- A 450 Page book filled with sample programs, plus . . .
- A fast, standard, full featured C compiler that supports all data types and the latest features like bit fields, enumerations, structure assignment, and passing/returning structures.
- A fast linker that loads separately compiled files, searches libraries, and builds an executable program.
- An extensive library of more than 170 functions (including the standard C functions and the computer specific functions that provide direct access to the operating system and BIOS).
- Tools that allow you to optimize your programs for minimal space or maximum speed.

Operators are standing by . . . Please use this Number for ORDERS ONLY!

CALL TOLL FREE FOR RUSH ORDER DELIVERY!

1-800-523-9520

IN TEXAS, PLEASE
CALL TOLL FREE
1-800-622-4070

For Technical Support Please call 1-214-783-6001

MIX Software, Inc. / 2116 E. Arapaho / Suite 363 / Richardson, Texas 75081

Or contact our Worldwide Distributors direct in:

Canada: Saraguay Software 1-416-923-1500 Switzerland: DMB Communication CH-1-825-53-29
Australia: Techflow 047-586924 France: Info/Tech 1-43-44-06-48

RUSH REPLY ORDER FORM!

Please check method of payment:

Check Money Order MasterCard/VISA

Your Card #: _____

Expires _____

Shipping Charges: (No charge for ASM Utility)

In the U.S.A.: Add \$5.00 per Order.

In CANADA: Add \$10.00 per Order.

OVERSEAS: Add \$10.00 per Text Editor. Add \$20.00 per C Compiler. Add \$30.00 for combined C Compiler and Text Editor.

Operating System: (Check one)

CP/M Z80 MSDOS/PCDOS

Specify Your Computer Name _____

Specify Disk Format _____

NAME _____

Telephone A/C (_____) _____

Street _____

City _____

State _____

Country _____ ZIP _____

MIX
software

2116 East Arapaho

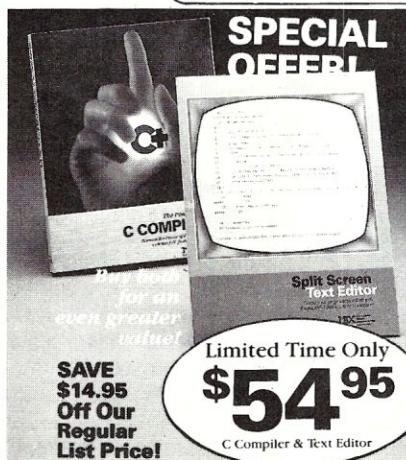
Suite 363

Richardson, Texas, 75081

Ask about our Volume Discounts!

Call 1-214-783-6001

M



Description

Quantity

PRICE

Total

Split-Screen Text Editor

\$29.95

\$_____

C Compiler

\$39.95

\$_____

C and Text Editor (Special)

\$54.95

\$_____

ASM Utility

\$10.00

\$_____

Texas Residents Add 6.125%

\$_____

Sales TAX

\$_____

Shipping Charges (See at Right)

\$_____

TOTAL OF YOUR ORDER:

\$_____

For the Advanced Computer User

Micro/Systems Journal™

September/October 1986
Vol. 2 No. 5

STAFF

Publishers
Sol & Lennie Libes

Editors
Sol & Don Libes

Associate Editor
Susan Libes

Contributing Editors
*A.G.W. Cameron
Robert Blum
Ian Darwin
Stephen R. Davis
Dave Hardy
Ken Jackson
Henry Kee
Steve Leon
William Wong*

Editorial Assistant
Lenore Melillo

Advertising
Lennie Libes

Circulation/Administration
*Lennie Libes
Linda Gohlke*

SUBSCRIPTION RATES	1 Yr.	2 Yrs.
U.S.	\$20	\$35
Canada & Mexico	\$28	\$52
Europe & So. America	\$35	\$64
Asia, Africa, Australia & Near East	\$39	\$70

ADVERTISING RATES: Available on request. Call (201) 522-9347 or write to: M/SJ, Box 1192, Mountainside, NJ 07092.

CHANGE OF ADDRESS: Please send old label and new address to: Micro/Systems Journal, Box 1192, Mountainside, NJ 07092.

CORRESPONDENCE: Address all letters to: M/SJ, Box 1192, Mountainside, NJ 07092. The editor may also be contacted via MCI Mail (SLIBES).

Micro/Systems Journal (ISSN #8750-9482) is published bimonthly by Libes Inc., 995 Chimney Ridge, Springfield NJ 07081. Second Class postage is paid at Springfield, NJ and at additional offices. **POSTMASTER:** send address changes to MICRO/SYSTEMS JOURNAL, PO Box 1192, Mountainside NJ 07092.

Copyright © Micro/Systems Journal, a subsidiary of Libes Inc. All rights reserved, reproduction prohibited without permission.

Micro/Systems Journal is a trademark of Libes, Inc.

IN THIS ISSUE

FEATURE ARTICLES

Control Systems Made Simple	20
<i>Lynwood H. Wilson</i>	
Enhancing CP/M-80	28
<i>William F. Dudley</i>	
A First Look At The 80386	34
<i>Michael Weinreich</i>	
Interfacing Using The SCSI Bus	46
<i>Hul Tytus</i>	
Program Interfacing To MS-DOS - Part IX	62
<i>William G. Wong</i>	
Using SYMDEB With NMI Breakpoints	66
<i>James G. Owen</i>	
SYSLIB, Z3LIB & VLIB	72
<i>Richard Conn</i>	

PRODUCT REVIEWS

PC-Pro	38
<i>Alex K. H. Soya</i>	
CompuMagic Utility Package	80
<i>Joseph A. Sabin Jr.</i>	
Plu*Perfect Systems' DataStamper	84
<i>Gord Wiggins</i>	

DEPARTMENTS

Editor's Page	4
News, Views & Gossip	6
<i>Sol Libes</i>	
Mail	10
Turbo Pascal Corner	14
<i>Stephen R. Davis</i>	
The C Forum	16
<i>Don Libes</i>	
The PC/Blue Report	78
<i>Hank Kee</i>	
The SIG/M Public Domain	82
<i>Steve Leon</i>	
The Unix File	88
<i>Ian F. Darwin</i>	
Software Directory	91
New Products	92

AUTHORS: Micro/Systems Journal is always seeking good articles. Please write or call first to see if we are interested in the subject. Please do not send the article unless we ask for it.

If you are interested in reviewing hardware or software please write telling us your interests, your background and include a sample of your writings.

Send a stamped self-addressed business size envelope for a copy of our Author's Guide.

TRADEMARK ACKNOWLEDGEMENTS

Macintosh	Apple Computer Company
UNIX	AT&T
GEM, CP/M	Digital Research Inc.
Concurrent DOS	Digital Research Inc.
8088, 8086, 80286	Intel Corporation
IBM PC, PC/XT, AT	International Business Machines
MS-DOS, XENIX	Microsoft
GW Basic	Microsoft
68000, 68010, 68020	Motorola

Editor's Page

GUEST EDITORIAL by Al Levy



I assemble and sell PC clones. As an OEM dealer I have learned much of the trials and tribulations of dealing with clone component suppliers. (You think you have problems?)

First, one must learn to deal with manufacturers and distributors without developing ulcers. To my knowledge, the electronics industry is unique! When you are promised delivery on Wednesday, sure enough the shipment arrives on Wednesday. Unfortunately nobody remembers to tell you WHICH Wednesday.

In the music industry, arriving late once meant no more opportunities! If you didn't deliver the goods that was also your last chance.

Finally, all the pieces have arrived. Now for the experience of assembling a computer. You unwrap the boxes, and there are quite a few per machine. Let's count them:

- 1) Case
- 2) Power supply
- 3) Mother board
- 4) Disk drives (usually 2)
- 5) Video card
- 6) Disk controller card
- 7) Hard disk controller card
- 8) Hard disk
- 9) IO card
- 10) 256K memory chips
- 11) 64K memory chips
- 12) Speaker
- 13) Monitor
- 14) Keyboard

The case looks OK except for a scratch on the back. Maybe your customer, who has been calling daily, won't notice. The keyboard looks fine, except it has two 6-keys in the numeric pad and no 5-key. The drives are not the ones you ordered because XYZ company isn't shipping this month. You have the distributor's assurance that the drives you received are "every bit as good as....". Then you notice that the memory chips are KOREAN at 200ns when you ordered JAPANESE at 150 or 200ns. Like "H E L P!".....You make some phone calls and pray.

When all has been straightened out, you begin assembly. It feels like the birth

of a child, organizing this pile of inert electronic junk and beginning the creation of life(?). One by one you casually pop the 36 memory chips into place. In show business, "break a leg" brings good luck. In the computer-biz break-a-pin means a dead bug, or loosely translated, a broken chip. When this heroic feat is done, pat yourself on the back and think, "great, the hard part is over"

Experiences of a PC Clone Dealer

Next, install the little gold posts that hold the mother board to the case. This requires nine posts, (count 'em, nine lovely posts), and eighteen insulated washers. (Total cost about 50 cents.) Terrible! You sent those clowns a few thousand dollars and received seven gold posts and three washers. So, run to the local hardware store (muttering curses and swearing never to do business with XYZ again), and buy the needed parts.

Now to screw the mother board into the case. OOPS, the holes don't line up. Almost, but no cigar. Get out the drill.

Mother board now installed. Mission impossible almost accomplished. The power supply is next, (no problem). Now the time for groaning as you press a quarter inch piece of plastic into a space one eighth inch wide. You succeed the first time, except the wires are all backwards.

The moment of truth is approaching. The video card pops in nicely. Carefully remove the monitor from its box and attach it to the computer. The AC plugs go into the wall sockets. If everything works you should see a memory check. Bravely turn the off/on switch on and voila.....no video. Your response is immediate, quick thinking at its best. Turn the machine off.

Right! You forgot to set the switches on the mother board. Not trusting your memory, you check the mother board manual and set the switches. The computer is switched on again and "hooray," it's working. (Did the customer order amber or green?) You know the answer, whatever you haven't received.

Did you ever look inside a PC? Check the drive cables....it's a jungle in there. Why do they make cables that have to be bent backwards to make the top drive the <A> drive? Screwing the floppies and hard disk into the system is a study in itself. It comes pretty easily if you are by nature a pretzel.

The cards are put into the slots and the big minute arrives. A DOS disk is placed in the A-drive and the system boots. Now the debug ritual for setting up the hard disk. Only problem, the fella who shipped out the hard disk forgot to enclose the spec sheet. This problem can't wait. A call is placed to the tech, who's out to lunch. The promise is made that he will call back by two o'clock California time...nobody tells you which day!

Once passed the sub/formatting, we invoke the great god FDISK who partitions our fixed storage media. Formatting the hard disk and creating a few sub-directories for your customer is S.O.P. Everything works well. Button up the machine and call the customer with the news, "your computer is ready." Everyone is excited.

Mr. Customer comes to pick up his computer and exclaims in delight "you know, it flickers."

I hope you realize that much of this is tongue in cheek. What I am earnestly stating is, details don't matter to the clone component makers. The consumer, or the dealer, becomes the testing ground. In the best situations, the dealer swallows all the grief prior to delivering the equipment. Occasionally, bad things happen later. It is usual for a retailer to test equipment from 5-10 days to prove it out. Don't break his chops. More often than not, he is protecting you. [u]

SEARCH 3.0

dBASE III

- A GENERAL-PURPOSE CATALOGUE SYSTEM WITH FAST, MULTIKEY BOOLEAN SEARCH -

- User abstracts **keywords** from source materials:

REAL ESTATE LISTINGS

LEGAL TRANSCRIPTS AND DEPOSITIONS

COMPANY AND CLIENT PROFILES

PERSONNEL PROFILES

RESUME'S

PATIENT CHARTS

ARTICLES, BOOKS, CLIPPINGS

WRITER'S OR RESEARCHER'S NOTES

HOLIDAY TOURS AND TRIPS

— Stores the abstracts in a simple universal file structure.

— Searches the "library" for relevant abstracts using **general multikey boolean forms** composed of **lists of keywords**.

- List syntax for boolean forms is simple; lists are stored to file for easy editing.

• **Search algorithm is FAST.**

• **Every application handled in the same way.**

- The all-purpose file structure has just two fields: abstract id number, and keyword. One record for every keyword. User sets the widths of the fields.

• **Categorization of keywords** is accomplished by a variable one-character prefix on the keyword (any printable ASCII character).

- Search lists come in four types: MATCH, NONMATCH, GRTR/= and LESS/=.

• Multilist queries come in two types: ALL and ANY. An item satisfies an ALL query if it satisfies all the lists (AND logic). An item satisfies an ANY query if it satisfies any list (OR logic).

- A list may have any number of keywords, and any mix of categories.

• **Search output is a list of "hits",** the id numbers of the abstracts which satisfy the query, stored to a file.

Resume Example

Categories are: T = title, F = function, G = year graduated, P = product area, \$ = salary, D = degree, M = major, S = school

Sample Abstract #43	Sample Queries	
id keyword 43 T_Asst. Director 43 F_Sales 43 G_73 43 P_Fabrics 43 P_Yarns 43 P_Clothing 43 \$_37000 43 D_MBA 43 M_Marketing 43 S_Columbia U.	(a) experience in Sales and Marketing; not a V.P. or Director; 10-15 years work experience; current salary not higher than \$50K. ALL MATCH F_Sales MATCH F_Marketing NONMATCH T_Director T_V.P. GRTR/= G_71 LESS/= G_76 LESS/= \$_50000	(b) not a steel industry executive (Director, V.P. or C.E.O.) ANY NONMATCH T_Director T_V.P. T_C.E.O. NONMATCH P_Steel

- Searching is fast because all entries across all categories are indexed simultaneously — all "search keys" are inverted at once.
- The system is menu-driven except for file creation and editing.
- Users should know dBASE III, but don't need to be programmers.
- dBASE III source code is included, so if you are a programmer you can adapt the system to taste.
- The system provides an option for automatic encoding of keywords and phrases into 3 or fewer bytes, leading to dramatic disk savings. (An application of this type is included with every order: abstracts of dBASE III material from TechNotes.)

To order, please send a check for \$49 + \$3.50 (S&H) to

Thinker's Apprentice / 392 Central Park West, Apt. 12X / New York, NY 10025
(212) 222-5050

(Residents of New York State, please add state and local sales tax.)

dBASE III is a registered trademark, and TechNotes is a copyright publication of Ashton-Tate.

Newn' Views & Gossip

by Sol Libes

RUMORS & GOSSIP

At a recent meeting of the Wall Street Security Analysts, IBM publicly acknowledged that the PC is facing increasingly stiff competition from the clone market. IBM's Chairman of the Board, John Akers, said that IBM plans to respond by increasing its PC products and services. He warned that if this segment of the computer industry becomes "increasingly commodity like," with price as the controlling factor, "you will probably see the IBM company departing from those parts. We are in the business of high-margin sales."

According to reports from market-research firms, IBM's share of the PC-compatible retail market is now less than 40%, down from 55% in January of '85. IBM's leading competitors in this market are: Compaq, AT&T, Leading Edge, Epson, Zenith, Kaypro, and several Asian companies.

Shipments of IBM's new RT RISC desktop systems are reported to be slightly ahead of expectations, according to an unidentified IBM executive. Projections are for 8000 systems to have been shipped by year's end.

Intel Corp. is rumored to be developing an 80386-based motherboard to replace the AT's 80286-based motherboard. (See the last part of this article for more on the 80386.) Also expected is a PC card with Intel's new video chip for high-resolution graphics.

Commodore International has decided not to introduce into the U.S., two PC-compatible machines currently marketed in Canada and Europe. Rumors are that they need to move a huge inventory of Amiga machines first.

The number of exhibitors at this year's **National Computer Conference** (once considered "THE" computer show of the year) dropped by over 20% from last year. With attendance down as well, the decrease in participants has been ascribed to computer conference "burnout" (there had been five major shows in the two months just prior to NCC '86), as well as competition from highly-specialized shows.

In the first quarter of '87, Microsoft is expected to release a new version of Windows which will operate under MS-DOS version 5. In addition to improved performance under DOS 5.0, new features, such as mouse macro's, will be included. There are also rumors that Microsoft will introduce its

own super-high resolution, color-display controller card for the PC based on the TI-34010 chip. The unit should provide 640 by 480 pixels (the EGA can give only 640 by 350) and will be much faster than the EGA card.

Dataquest, a market-research company, estimated that world-wide sales of PC-compatibles were \$7.1 billion in 1985. This is predicted to rise to \$9.5 billion in 1986.

It is rumored that Exxon is seeking to divest itself of its Zilog subsidiary and that AT&T and Honeywell are interested. Zilog, along with a dozen other companies, was bought during Exxon's high-tech buying binge fueled by the bullish oil market of the '70's. Most of these acquisitions have since been closed or sold. The pending sale of Zilog is ironic as the company has just recently gotten out of the red-ink area.

While Apple quietly dropped its plug-in LCD display for the IIC, IBM noisily introduced its laptop, the PC Convertible, which has a plug-in LCD display. Apple will continue to market a CRT display for the IIC. Future offerings from Apple include videodisk and CD-ROM products designed for school use. The National Geographic Society and Lucasfilm are helping Apple put together the digitized sound, graphics, and story segments. The disks are expected to operate on a new version of the Apple-II that will have a high-resolution display and the 65816, 16-bit processor. An upgrade will be offered for existing Apple-II systems.

Speaking of Apple, it's been reported that MacDonalds Corp. is suing all Macintosh product developers (including Apple) over the use of the term "Mac" as a modifier in any product name, e.g. MacDraw, MacWrite, MacTutor.

George Morrow, who got his start making S-100 boards, is back in business. His new enterprise (Intelligent Access, Oakland, CA) will be producing PC-compatible boards.

3Com and Codenoll Technology have joined forces to bring out a version of the popular 3Com Ethernet network which operates over a fiber-optic cable.

IN THE PUBLIC DOMAIN

PC-SIG (Sunnyvale, CA), a commercial distributor of PC/MS-DOS public-domain software, is now distributing their entire software library on a single Compact Disk (CD-ROM) for \$195. The library was

previously available as a set of over 500 floppy disks costing over \$2000. PC-SIG is also selling the Hitachi CD-ROM reader drive with a PC-interface card for \$995 (comparable CD-ROM readers with interface now sell for as little as \$600). A menu-driven program is provided on-disk to assist in locating programs.

If you buy a CD-ROM and reader from Reference Technology Inc. (Boulder, CO), they'll give you a disk containing the PC-SIG library.

IBM EGA-II EXPECTED SOON

IBM recently released, at least to software developers, the specifications for the new version of their Extended Graphics Adapter (EGA) card, called the "EGA-II". The EGA is rapidly becoming the display standard for PC-compatible systems.

When IBM introduced the EGA, it got a cool reception. Its basic problems were high cost (almost \$1500 for the card and display) and the lack of software that could take advantage of its features. It took only about 9 months for the clone board makers to introduce EGA-compatibles, for the display makers to introduce EGA-compatible displays, and for the software developers to add EGA drivers to their products. The competition has resulted in the EGA/display combo dropping to under \$800 in price and becoming the new standard for high-resolution, color graphics. In addition, many of the clone boards offer features (e.g. Hercules emulation) surpassing the performance of IBM's EGA. This has caused a significant drop in IBM's share of the EGA market.

Now comes word that IBM's EGA-II will have a raft of new features including enhanced resolution, more modes, more colors, and an increased capacity to download fonts. It is also expected to require an analog display, whereas the current breed of EGA uses a digital display. The cost of the card and display will be in the \$1500 to \$2000 range.

I give the EGA clone makers about 6 to 9 months to catch up with this new IBM product, driving the price down under \$1000. It will probably take a little longer for the software houses to meet the challenge. In the meantime, I expect the price of the EGA-I/display combination to drop below \$600.

AT COMPATIBLES TAKE THE LEAD

Less than two years ago, IBM began shipping their AT machines. It is estimated that well over 500,000 AT's have already been sold. This has partially compensated IBM for the loss of its share of the PC-XT market. According to Future Computing, a market-research firm, the AT now accounts for 32% of IBM's desktop sales and will probably soon surpass their XT sales. Nevertheless, IBM has not achieved the market share with the AT that it once had

with the PC-XT's. And, although IBM is still the market leader, its share of the AT-class market is eroding rapidly. It is estimated that over 1.2 billion AT-class machines will be sold this year, with IBM responsible for only 45% of the sales. In addition, the Japanese, Koreans, and Taiwanese have only just recently begun to ship their versions of the AT. As a result, the single-unit price of a basic AT machine is already under \$1500, with additional discounts for large-volume purchases. By this time next year, IBM's market share will probably be even less as AT-clone prices drop further.

There are already almost 40 AT-compatible vendors. Compaq, KayPro, NCR, HP, AT&T, Tandy, Wang, Sperry, and TI (listed in order of decreasing volume) are IBM's leading AT competitors. Virtually all of the clones have features not available on IBM's AT. It is surprising to find KayPro in the number three position, ahead of such heavy weights as HP, AT&T, and Tandy. By next year, the number of AT-compatible vendors will be even larger and the Far East suppliers will probably be major forces in the market.

Prices for the 8-Mhz AT knockoffs equipped with 30-Mbyte hard disks are already nearing the \$2000 level. IBM can be expected to cut prices to meet this competition. As the price gap between the XT and AT class machines is quickly narrowing, the AT market should continue to increase as PC-XT owners trade in their machines. The introduction of DOS 5.0 next year should also spur sales, as this version of DOS requires a 286-based system.

IBM LAPTOP ANOTHER LOSER?

IBM's new laptop machine, called the "PC Convertible", appears to be living up to one suggested interpretation of IBM's acronym, i.e. "Inferior But Marketable." But then again, it may not be living up to the marketable part.

Its poor showing thus far is apparently due to IBM's late entrance into this emerging market with a product that has fewer features, lower performance, bigger size, and higher price than its competition. Unlike well-designed laptops, such as the Toshiba 1100, the PC-Convertible is unable to run some of the most popular programs, e.g. LOTUS 1-2-3. In addition, its built-in modem is not Hayes compatible and therefore will not work with most of the popular communications programs. The LCD display also leaves much to be desired. With the plug-in options attached, the size and weight are increased such that the unit can no longer be considered a laptop. There is no 5 $\frac{1}{4}$ " disk option, so users must add an expensive 3 $\frac{1}{2}$ " disk drive to their PC or AT. Adding in the cost of the options, the price of the PC-convertible is also substantially higher than that of its competition. Those same features are sold as standard equipment in the Toshiba and other laptops. Furthermore, high-performance 286-based laptops are already available.

SCSI STANDARD APPROVED

ANSI (American National Standards Institute) has given final approval to the SCSI standard (Small Computer Systems Interface). The SCSI is a high-performance bus designed to direct the use of several intelligent peripherals on a single CPU. If the bus is accepted by peripheral and system manufacturers, it will ease development of hard-disk, tape, and optical-disk drives, as well as other peripherals. (See the article in this issue.)

There are already a large number of products equipped with the SCSI, unfortunately some were made before ANSI began work on a set of common commands for controlling SCSI-interfaced, hard disks. This upgrade, which will be called "SCSI-II", is expected to be released in the next year.

Apple has announced that it will include an SCSI interface on its new version of the Macintosh. DEC, AT&T, Wang, Sun Microsystems, Hitachi, as well as several others, are also supporting the new standard. Though IBM has not as yet declared its intentions, there are rumors that the company plans to use the SCSI for a CD-ROM drive to be marketed by late next year. If IBM ignores the SCSI and goes its own way, this could eventually kill the standard.

ARE SOFTWARE WARRANTEES REALLY WARRANTEES?

Doesn't it bug you to read the Software Warranty on a typical commercial software package? The supplier usually states that it "makes no warranty with respect to this software" and that the product is sold "as is" with "the licensee assuming the full risk as to quality and performance." When you get down to it, this is no warranty at all!

Hopefully, this may soon change as ADAPSO, the industry association, is applying pressure to several of the larger software suppliers. They are proposing that the software makers offer a refund within 90 days for programs found to have a bug or if the product does not meet the maker's ad claims or specifications. According to ADAPSO MicroPro, Lotus and Ashton-Tate have already agreed to this.

200-Mhz RISC CHIP SET BEING DEVELOPED

DARPA (Defense Advanced Research Projects Agency) is funding the development of the first gallium-arsenide, 32-bit, RISC (Reduced Instruction Set Computer), chip set. Contractors for the project are Texas Instruments and McDonnell Douglas with delivery slated for mid-'88. The chip set is expected to operate at a 200-Mhz clock rate, executing most instructions within one clock cycle. Lower-performance

CMOS versions are expected to be made available as commercial products. Also funded was the development of an ADA compiler, Common LISP, and MACH (a new high-performance operating system compatible with UNIX).

AT CLONE COMPATABILITY PROBLEMS

Some users are reporting compatibility problems with some AT clones. The primary problems occur when these units are used in networking environments. Units that operate at 10 and 12Mhz with no wait states and are used with the IBM token Ring Network and Novell system appear to be among those with problems. In some cases the units simply do not work and in others have unexplainable crashes.

AMIGA AFTER ONE YEAR

The Commodore Amiga will be one year old this October. Commodore claims that by its first birthday it will have shipped 110,000 machines but industry pundits think it will be more like 30,000 units. The unit, which is actually made by Sanyo, is carving out a niche among users who do video production and simulation in real time, or near real time. It has achieved quite a following in the TV broadcasting business. Graphics designers can easily synchronize, or genlock, external video sources to the machine.

However, in the larger personal computer marketplace its sales have proven a disappointment. Despite substantial price cuts (it now sells for under \$1,000) it has not been able to compete against systems of lesser performance, such as the Macintosh, PC and Atari ST. There is no doubt that Commodore is yet to turn a profit with the system.

A second-generation Amiga, based on the 68020, with higher resolution and an internal 20Mbyte drive is expected to be introduced by year-end. However, Atari and Apple are also expected to introduce 68020 systems. All of these systems are expected to sell in the \$3,000 range, allow execution of PC software and be usable as workstations on a networking system.

THE PC/MS-DOS/UNIX/80386 CONNECTION

With its virtual architecture, the 80386 will permit applications to be run concurrently under different types and copies of operating systems simultaneously. Three companies have already announced plans to release the first operating system for 80386-based machines. If IBM drags its feet in introducing its 386 system, one of those three just might beat them.

Interactive Systems has a contract with Intel Corp. and AT&T-IS to port the UNIX operating system to the 386. Interactive Systems has teamed up with Phoenix Software Associates to develop V-86/IX, UNIX

Does this look familiar?
What if each change
you made to your
program was ready to
test in seconds instead
of minutes?



"The SLR tools will change the way you write code. I don't use anything else.", Joe Wright

RELOCATING MACRO ASSEMBLERS • Z80 • 8085 • HD64180

- Generates COM, Intel HEX, Microsoft REL, or SLR REL
- Intel macro facility
- All M80 pseudo ops
- Multiple assemblies via command line or indirect command file
- Alternate user number search
- ZCPR3 and CP/M Plus error flag support, CP/M 2.2 submit abort
- Over 30 user configurable options
- Descriptive error messages
- XREF and Symbol tables
- 16 significant characters on labels (even externals)
- Time and Date in listing
- Nested conditionals and INCLUDE files
- Supports math on externals

requires Z80 CP/M compatible systems with at least 32K TPA

\$49.95

SLR Systems

1622 N. Main St., Butler, PA 16001
(412) 282-0864 (800) 833-3061

System V Release 3.0. The system will enable several MS-DOS and UNIX applications to run concurrently. The MS-DOS emulation will be developed by Phoenix. The system will be made available to other OEM's, with the single-user version promised by year's end and the multi-user version by mid-'87.

Several PC clone and workstation makers are expected to announce 80386-based machines. Look for them to run MS-DOS in the 8086 emulation mode or under UNIX V 3.0 using a PC/MS-DOS emulator. Many manufacturers are hoping that this bridge between UNIX and DOS will finally bring UNIX into the business office.

Locus Computer is developing a version of its Multisystem Merge for the 80386-based UNIX V 3.0. This software will run PC/MS-DOS applications as a task under UNIX. AT&T-IS had acquired Multisystem Merge and offered it to users to aid them in running DOS programs under UNIX on AT&T's 6300-Plus, 80286-based machines.

Softguard Systems, developer of a popular copy-protection system, has announced its VM/386 operating system which can run VM, UNIX, and DOS applications concurrently. And, Award Software (Los Gatos, CA), a DOS BIOS supplier, is promising a 386-based ROM BIOS.

Intel is boasting that it will ship over 25,000 of the 386 chips in the third quarter of this year, with over 100,000 by year's end. This means that Intel is ramping up

production of this chip at a much faster rate than was done with the 8088 or 80286 chips. If this is the case, 386-based systems may dominate the personal computer market by the end of the decade. In comparison, Motorola has shipped about 90,000 of the 68020 chips so far.

COMPUTER MAGAZINE BIZ UPDATE

Last year was a terrible year for computer magazines (in spite of it we started M/SJ). Two of the top sellers (Creative Computing from Ziff-Davis and Popular Computing from McGraw-Hill) closed their doors. Circulation of the top-ten magazines dropped 6%, the number of ad pages dropped 20-25% and ad income dropped 10-20%. Since more magazines closed up than started there has been an overall decrease in the number of computer magazines published. As a large number of the remaining magazines are no longer profitable, we will probably see more close this year.

There has also been a shift in position. For example, BYTE magazine, the long-time circulation leader, no longer ranks in the top ten. Who is number one, you ask? Why it's Family Computing, published by Scholastic's Inc., with a circulation of 610,000. Second is Personal Computing, followed by Lotus, PC Magazine, and Run.

HINT OF THE MONTH

If you have an AST PC multifunction card and need to replace the clock-calendar

BOBCAT

The finest disk catalog program available
for PC-DOS or CP/M

- * Keeps track of files on floppies or hard disks
- * Compatible with PC-DOS 2.0 or 3.0; CP/M 2.2 or 3.0
- * Contains the most advanced features:
 - 65 character individual file comment line
 - 45 character individual disk title
 - ASCII BOBFILE (disk number file) storing the disk title and file comment lines. Great for quick finds or looks in each disk or directory
 - full directory and subdirectories compatibility
- * File comment lines for both text and binary files
- * Unprotected for ease of use
- * A proven utility sold since 1981

Price: US Residents \$49.95 US funds
Canadian Residents \$49.95 Canadian funds
Other Countries \$54.95 US funds
plus \$3.00 postage and handling
Ontario Residents please add 7% sales tax
Extra DOS or licenses - \$15.00 each to \$75.00 maximum



MASTERCARD, VISA, BANK DRAFTS, MONEY ORDERS
COMPANY CHECKS, PURCHASE ORDERS
Please allow 10 days clearance for PERSONAL CHECKS

R&L MicroServices Inc.
Box 15955, Station F
Ottawa, Ontario, Canada
K2C 3S8 (613)225-7904



The home of the BOBCAT

battery (Matsushita BR2325) you will discover it is not a standard stock battery and is only available from AST for \$10 plus shipping. But don't fear, a Duracell Lithium DL2025 fits the holder, provides the same voltage, and is readily available in most parts stores at considerably less cost. Though slightly smaller in diameter than the original battery, it fits quite well in the holder.

NOTEWORTHY NEWSLETTERS & MAGAZINES

If you are into ZCPR3, check out the **Z-NEWS newsletter**, published every other week. It is devoted to supporting ZCPR3 and Z-System users. A years subscription is \$24 or send \$1 for a sample issue. Echelon Inc., 101 First St, Los Altos CA 94022.

Bruce Morgen, author of the *Advanced User* column in USER'S GUIDE magazine has started a newsletter for 8-bit system users called the **NAOG Newsletter** (North American One-Eighty Group). Cost is \$15 a year for 12 issues. NAOG, Box 2781, Warminster PA 18974, (215)443-9031.

The **Computer Journal** is devoted to those who interface, build and apply micros. It appears to be oriented to S-100 and Apple-II users. A sample copy is \$3.25; subscriptions are \$14/yr and \$24/2yrs and is published bimonthly. The Computer Journal, 190 Sullivan Crossroad, Columbia Falls, MT 59912. P

DRIVE INN

Enclosure & power supplies for
FLOPPY, WINCHESTER, TAPE DRIVES, SINGLE BOARD COMPUTERS & S-100 SYSTEMS

8 inch
5 inch
3 inch

Call or write for free catalogs & application assistance

— INTEGRAND

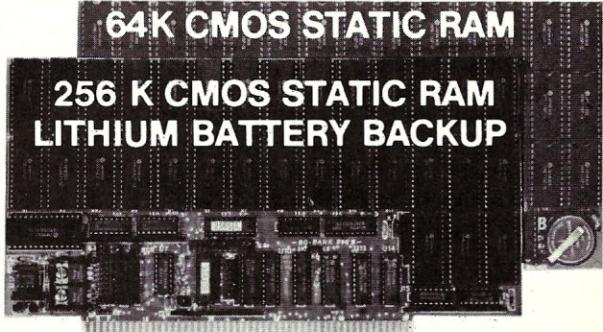
RESEARCH CORPORATION

8620 Roosevelt Ave. • Visalia, CA 93291
209/651-1203

We accept BankAmericard/Visa and MasterCharge



HIGH PERFORMANCE RAM



✓ COMPARE

	8/16 BIT DATA	Bank Selection MPM, OASIS, CROMIX-D	10 Megahertz Speed	Software Write Protect	Battery Backup Option
CompuPro Ram 22	✓	NO	✓	NO	NO
Octagon 256K	✓	NO	✓	NO	NO
Cromemco 256KZ II	✓	NO	NO	NO	NO
Dynamic Boards	✓	✓	NO	NO	NO
BG-Bank 256S	✓	✓	✓	✓	✓

GUARANTEED IN YOUR SYSTEM

CROMIX-D • MPM • CCS • OASIS • AMOS

✓ PLUS: 8/16 BIT TRANSFERS • 24-BIT EX. ADDRESSING
8-12 MHZ • 2K DESELECTS • RAM-EPROM MIX
IEEE 696/S-100 • LOW POWER • FULLY STATIC

LITHIUM BATTERY BACKUP avoids power failure crashes intelligently. Unique POWER-FAIL-SENSE circuit allows processor to save register information and disable board before POWER FAILURE CRASHES memory.

BG BANK 256S..... \$695 Battery Backup \$79
BG BANK 64S..... \$299 ~~Battery Backup~~ \$59

call for new lower prices

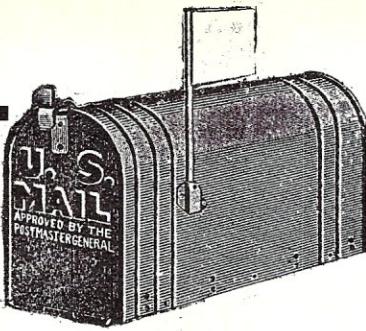
MEGABYTE MADNESS
FOUR 256S CARDS

~~\$299.50~~

call for new lower prices

BG COMPUTER APPLICATIONS, 206 Brookside,
Bryan, Texas 77801. International orders add 30%.
(409) 775-5009

there is mail....



We welcome your letters with comments, compliments, criticism and suggestions. We read them all and publish the most noteworthy, even if they are critical of us. We do not have the staff to answer all letters personally. And all letters become the property of M/SJ and may be subject to editing. Further, we do not print letters that do not include a name and address.

Please send your letters to: Micro/Systems Journal, Box 1192, Mountainside NJ 07092.

LIKES THE M/SJ EDITORIAL APPROACH

Dear Libes:

Thank you for continuing publishing the magazine which shows the best mix of articles on CP/M and PC/MS-DOS. Your editorial wisdom shines over several magazines I subscribe to. As an owner of both 8-bit and IBM PC/XT compatible machines, I straddle across information gap. Micro/Systems Journal is the one I use to go back and forth between the two worlds.

As a computer user (never an innovative one) who cut his teeth with CP/M, I deplore wholesale abandonment of CP/M world, but at the same time, I feel that our eyes should be on the future of the microcomputer world which appears to be moving toward multi-user, multi-tasking system of UNIX, Concurrent DOS, and MS-DOS. I am a type who feels very sad whenever I use MS-DOS version of WordStar 3.31. Being of CP/M heritage, WordStar 3.31 is *not* programmed to use a full capability (especially graphics) of MS-DOS. Because of CP/M constraints which are still visible in WordStar 3.31, WordStar lost its productivity edge to newcomers, such as Microsoft Word, and Word Perfect. I am happy to learn that MicroPro is coming up with WordStar II, which I suppose uses all of the MS-DOS advantages.

I have checked software which has both CP/M and MS-DOS versions, such as Turbo Pascal, Condor (database management program), dBASE II, Crosstalk, MITE, Milestone, SmartKey II, SuperCalc, and others. There are some improvements in productivity for the MS-DOS version, but they are not as impressive as the software written from scratch for MS-DOS. I wonder why. The only exception is dBASE which seems to be written fresh from the start (keeping, however, most of CP/M version's command structure) for MS-DOS.

In July/August issue, Steve Leon told us about his wishes for the types of CP/M software he wants to see. I really like to see along with Steve, "a way (will be found) to

run a CP/M machine off PC serial port as a second console for the PC..." I really like Steve's thinking on how to use an 8-bit machine, perhaps 3-4 years old. These 8-bit machines will be worth only a few hundred dollars now, and they are destined to a junk pile *without our tender, loving care*. As a part of TLC to my 8-bit machine, I have installed R. Conn's Z-system, and I added RAM disk. I really appreciate reading articles in your magazine on how to use a 8-bit computer, once you get your own IBM PC clone!!

Please don't hesitate to include as many articles on 16-, 32-bit microcomputers in the magazine, as long as you have a few solid CP/M articles.

Sincerely yours,
Paul Naitoh
San Diego, CA

MAGNUM DIGITAL REVIEW CORRECTION

Dear Sol,

Dion Vaughan of Magnum Digital, Inc has pointed out an error in my review of the Pro-189 CPU and FD-100 floppy disk controller. I stated that Magnum Digital implemented the console on Port 1 using handshake lines, while the second serial port did not have handshake lines implemented. I was wrong; the case is just the reverse. The serial handshaking is implemented on the spare serial port and *not* on the console port, which is as it should be. I apologize to Mr. Vaughan and to Magnum Digital for this negligence.

Mr. Vaughan also informed me that Magnum Digital now supplies ROMs for CompuPro (Viasyn) and Morrow controllers. Users can now install a Pro-180 board and use their own controllers. Further, the monitor has been improved and now incorporates features which were only available using the separate CTRL.COM. Direct boot from disk is supported on CompuPro and Morrow systems. Lastly, ZCPR3 is fully implemented and available with the purchase of the Pro-180.

I apologize for my errors and hope that this letter will serve to put the record straight.

Yours truly,
Steven D. Kaplin

MORE CONCURRENT-DOS ARTICLES WANTED

Dear Sol:

I was pleased to read the first installment of Alex Soya's articles on Concurrent PC-DOS. The article was factual, interesting, and provided information on an operating system which has not received much attention from the PRESS. I am looking forward to reading upcoming parts of this series as Alex continues to explore Concurrent and provide systems development information.

I have used several versions of Concurrent PC-DOS in a software development environment to develop software for industrial process control and instrumentation applications. Concurrent DOS has performed quite well, and has done most anything which I have asked of it (including such diverse requests as deleting the Terminal Management Processes, handling interrupt driven processes, and providing access to the shadow memory regions which store information for virtual consoles which are currently in the background).

DRI's documentation, however, allowed for much learning. I am hoping that Alex' experiences will ease that process.

Using an operating system that manages several virtual screens on one or two physical consoles (some projects have both a color and monochrome screen) has saved several months of development effort. It is designed as multitasking from the start, rather than for one of the commercially available *oil slicks* that attempts to convert a single tasking operating system into a multitasking environment.

Manx AZTEC-C86 provides an excellent development environment for Concurrent DOS. Their developer's systems allows a user to generate either .COM, .EXE, OR .CMD output files by simply setting a switch on the linker and using the correct run time library.

Jerry Jankura
Strongsville, Ohio

We are beginning to get the impression that DRI'S Concurrent-DOS is gaining in popularity with more advanced users. It is regrettable that DRI is not doing a proper job of supporting this product. Therefore, we expect to carry more articles, from Alex Soya, on Concurrent in future issues of M/SJ.

FROM AN ENTHUSIASTIC CONCURRENT USER

Dear Sir:

This letter is to applaud your recent article on the Concurrent PC-DOS operating system. I am a senior systems analyst and for the past two years have been involved in a project to develop a work station for the Air Force. We used the Concurrent PC-DOS operating system

A NEW ERA FOR HIGH PERFORMANCE TEST AND MEASUREMENT EQUIPMENT CONTROLLED BY PC'S

TURN YOUR PERSONAL COMPUTER INTO TEST AND MEASUREMENT EQUIPMENT. INSTANTLY!

There's been other hardware available, but if you really want to use it, you have to develop your own software. The others are just cards. You can't buy a piece of equipment, plug it in, slip in the disk and go to work. You're on your own.

Not with Rapid Systems.

Rapid Systems Test and Measurement PC Peripherals are ready to go, the minute you receive them.

Plug the Peripheral into your personal computer, slip in the disk, touch one button and you're ready to work. It's that simple and easy.

Plus, all the post processing capabilities of your personal computer are at your fingertips. Store and retrieve waveforms from disk. Analyze and process the information. Your computer provides intelligent control and analysis, and it's available for complex mathematical computations and word processing.

FINALLY, INEXPENSIVE, INTELLIGENT AND HIGH PERFORMANCE TEST AND MEASUREMENT EQUIPMENT.

Take a look at the performance and capabilities of our line of test and measurement PC peripherals. You'll discover technology and high performance at prices you wouldn't have thought possible. A Digital Oscilloscope Peripheral for only \$599. An Enhanced Scope for \$725. Spectrum Analyzer for \$788. Data Logger or Data Acquisition Peripherals—your choice at \$688.*

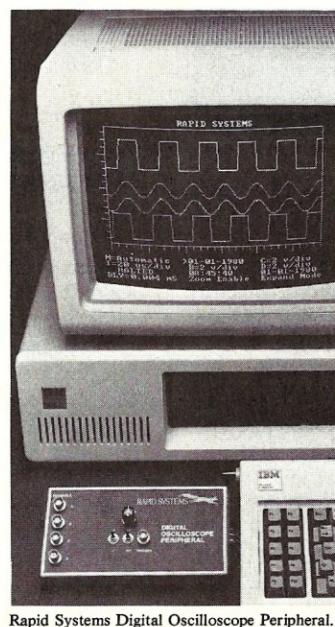
Powerful test and measurement equipment is as close as your personal computer for only a few hundred dollars.

DIGITAL OSCILLOSCOPE PERIPHERAL

For IBM, IBM compatible, Apple and Commodore personal computers. The peripheral hardware is contained in a compact 7.5" x 4" x 1.2" rugged ABS package that attaches to the computer via a ribbon cable and interface card. The digital oscilloscope software turns the personal computer into a versatile 4-channel digital oscilloscope.

Major features:

- Sampling rates to 2 MHZ.
- Variable gain: 1.6v to 320v p-p.
- Resolution to 6.25 mV.
- 8-bit A/D conversion.
- Simple menu driven turn key operation from the keyboard.
- Multiple mode triggering.
- Enhanced graphics display.
- Diode protected inputs to 300 v.
- Storage/Retrieval of waveforms.

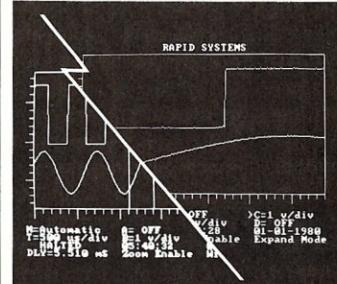


Rapid Systems Digital Oscilloscope Peripheral.

ENHANCED DIGITAL OSCILLOSCOPE PERIPHERAL

The enhanced digital oscilloscope is a software enhanced digital scope. The major features of this unit are:

- Zoom mode.
- Active scrolling of the waveform.
- Transfer data to 123/spreadsheets, dbIII files, or spectrum analyzer.
- Assembly language speed.
- Help file.
- Printer drivers.

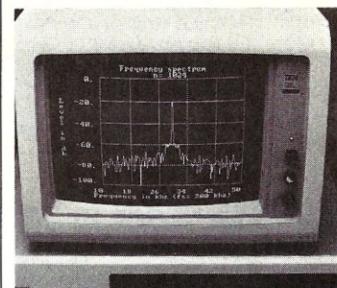


Zoom Mode of Enhanced Digital Oscilloscope Peripheral.

SPECTRUM ANALYZER PERIPHERAL

The Spectrum Analyzer Peripheral converts your PC into an instrument that characterizes an electrical signal for its frequency content. The peripheral uses the Fast Fourier Transform (FFT) in software to compute the signal spectrum. Major features:

- 2 MHZ digital scope software included free.
- Sampling rates to 500 kHz.
- Menu driven operation.
- Store/Retrieve/Print modes.
- Amplitude versus Frequency display using enhanced graphics.
- Hardware and Software triggering.
- FFT sizes: 16 to 1024 points.
- Frequency/amplitude cursor.
- Spectrum averaging.



Rapid Systems Spectrum Analyzer Peripheral.

DATA LOGGER PERIPHERAL

The Data Logger Peripheral allows the IBM or compatible personal computers to be used as versatile 4-channel data logging systems. The peripheral provides continuous measurement of up to four channels of data at a user selectable rate, and the data may be put out to hard disk or floppy. The number of data points to be taken is selectable by time or number of samples. Additionally, Start may be driven by either time and/or event (an internal or external trigger).

Rapid Systems Data Logger Help Menu	
(A)	F1 - Display this Help Menu (Alt+F1 for Page 2)
(F2	- Select Sample Rate ..Current Value is 1.0 Seconds/Single
(A)	F3 - Select & Display Channel 1 Gain ..Current Setting +/- 8.0 Volts
(A)	F4 - Select & Display Channel 2 Gain ..Current Setting +/- 40.0 Volts
(A)	F5 - Select & Display Channel 3 Gain ..Current Setting +/- 80.0 Volts
(A)	F6 - Select & Display Channel 4 Gain ..Current Setting +/- 160.0 Volts
(F7	- Convert Logged Data to Spreadsheet Format
(F8	- Convert Logged Data to Database Format
(F9	- Input Label for Screen Display
(F10	- Select Save Parameters or Redisplay Saved Data
(F11	- Select Function, CS, or Esc

Help menu from Data Logger Peripheral display.

Major features:

- 8-bit A/D conversion.
- Diode protection on all inputs.
- Sampling rates from .1 second to 999.9 seconds or event driven.
- Simple menu driven operation from the PC keyboard.
- Graphics display using 160x488 pixels for data display and data windows for parameter selection.
- Storage/retrieval of data on disk.
- Strip Chart printing with graphics printer.
- User selectable display headings.
- Automatic scrolling of a displayed waveform.
- Data reduction utility source and object code included!



Display from Data Acquisition Peripheral.

DATA ACQUISITION PERIPHERAL

The data acquisition peripheral allows the user to customize the PC for a specific data acquisition application. This unit provides a high speed, easy-to-use software interface for direct data access and control of the hardware. The User Definable Interface software (UDI) included with the peripheral provides assembly language subroutines dedicated to the control and acquisition of data from the hardware.

Major features:

- Digital oscilloscope software included free.
- Sampling rates to 2 MHZ.
- Data acquisition setup utility.
- Data transfer utility.
- Selectable gain/sampling rate.
- Trigger mode selection.

*Prices are for IBM compatible peripherals. Other models cost less. Additional software and accessories available.

For your copy of the new Rapid Systems Test and Measurement Peripherals catalog, to order, or for further information, call or write Rapid Systems, 755 N. Northlake Way, Seattle, WA 98103. (206) 547-8311. Telex: 265017UR.

RAPID SYSTEMS



Save**BUILD YOUR OWN IBM XT & IBM AT COMPATIBLE SYSTEMS****Save**

-Why Pay More-Build Your Own With Ease-Have Fun-Save a Fortune-
Introducing Super XT-16 Self-Assembly Kit

Assembled in Less Than 1 Hour with Screw Driver at SUPER LOW COST

- Including 640K XT-16 CPU Mother Board, 256 Installed, Color Graphic Card or TTL Monochrome Card, Floppy Disk Controller Card, One 1/2-height DS/DD Drive, Flip-Top Case, 135W Power Supply, Keyboard, Assembly Instruction and User's System Manual.
- We will assemble at no charge.
- Turbo Kit — Add \$50

ONLY
\$695.00

**XT, AT CASE**

- Same Dimension as IBM PC/AT
- For IBM PC/AT Compatible Mother Boards
- Flip-Top For Easy & Quick Access to Inside

ONLY \$60.00



\$119

XT, AT MOTHER BOARD

- XT-16 CPU Mother Board - \$167.00
- IBM PC/XT Fully Compatible, Run all Popular IBM Softwares
- 8088 Microprocessor w/8087 Optional
- 8 I/O Slots
- Up to 640K Memory on Mother Board
- Fully Assembled & Tested

AT-32 CPU MOTHER BOARD

- IBM PC/AT Fully Compatible
- 80286 Microprocessor w/80287 Optional
- 640K Standard, Upgrade to 1 MB on Board
- On Board Clock Calendar
- 8 I/O Slots

\$795

All Cards Fully Tested, Assembled & Warrantied / School & Institutional P.O. Accepted
OEM Dealers Welcome - Please call for our Special Dealer Prices

ATLAZ COMPUTER SUPPLY

616 Burnside Ave. / Inwood, NY 11696 • Mail Order Hotline 516-239-1854

(IBM is a trademark of International Business Machines Corporation)

because it provided such a rich set of professional operating system features rarely supported by other PC operating systems. I can also favorably compare Concurrent PC-DOS with UNIX since I have had 3½ years of professional experience with that operating system as well.

Of course, there are the legions of staunch PC-DOS users who will not take the time to learn about any other operating system even if that operating system provides nearly complete PC-DOS compatibility. Perhaps the ability to run multi-tasking PC-DOS applications does not appeal to them. Perhaps they would rather wait a couple more years for Micro-

soft to get it right. Or perhaps they should look at UNIX or (XENIX) and throw away their current PC-DOS software.

I can only say that this operating system has provided my project the ability to run nearly all PC-DOS software and provide background communications with a mainframe while providing foreground user functions for data base management and word processing. The Concurrent PC-DOS operating system has also provided a future for my project to grow to a multi-user/LAN cluster architecture with minimal effort.

Every operating system has its shortcomings, but for PC based systems de-

velopment, the Concurrent PC-DOS operating systems is a very fine product and deserves more exposure.

Again, I want to thank your magazine for such a fine article and I look forward to the future installments.

Sincerely,
Brian J. Mullan
Lutz, FL

MS-DOS CONFIG.SYS CAUTION

Dear M/SJ

Just a quick note re: "undocumented options" in the article on configuring MS-DOS (May/June 86).

Changing SWITCHAR value is much more dangerous than you indicate. Many programs assume the default value of "\\", and all kinds of things can fail in interesting ways. However, most dangerous of all are the BACKUP and RESTORE commands. BACKUP will happily make backups for you using whatever bizarre SWITCHAR you want. But when it comes time to restore anything, RESTORE goes down in flames if it comes across backup diskettes made with a nonstandard SWITCHAR. This is probably not what you want when you've lost a hard disk and your backup diskettes are the only thing between you and a nervous breakdown.

Tempting as it may be (especially for Unix people), it's best just to leave SWITCHAR alone. It's just not worth the possible problems later. Also, I've noticed that Microsoft people go into convulsions whenever the subject comes up, which does not bode well for possible future support of this option.

Otherwise, keep up the good work. I enjoy the magazine, especially articles on C, Unix, IBM. The Editor's page and the News, Views & Gossip column are very good also. I have zero interest in CP/M, so I ignore all such articles.

THANKS,
Dave Lutz
Canoga Park, CA

BACKUP POWER SUPPLY INTEREST

Sol,

I would very much appreciate your running an article on backup power supplies and/or the real utility of surge protectors.

There is so much hype regarding these gizmos, and they are bloody expensive! In the square wave vs. sine wave situation, the sine seems to be better from two standpoints: (1) area under sine wave is less than square wave, and thus is less likely to overheat the DC motor on hard disk drives, and (2) the voltage at ramp up and ramp down on a square wave is infinite, and some say that this condition can

continued on page 15

S100 USERS !!!

THUNDER/PLUS IS HERE 8087 ONBOARD

We at LOMAS DATA PRODUCTS never rest on previous success. The THUNDER 186 has been an overwhelming success over the past two and a half years but we just made it better. THUNDER/PLUS is now the most powerful sixteen bit single board computer on the S100 bus. We took all the features of the THUNDER 186 and added more memory and math coprocessor support and just for extra measure we increased the clock speed from 8Mhz to 10Mhz. This has been added to these additional features:

- 2 RS232 serial ports at 19.2 Kbaud max
- 1 Centronics compatible parallel port
- On board floppy disk controller will control both 8" and 5 1/4" disk drives including the new high density PC-AT disk drives
- 512K bytes of onboard memory standard, 1 full Megabyte of memory optional
- Up to 64K bytes of onboard EPROM
- Fully complies with the IEEE 696-1983 specification for the S100 bus
- Optional 8087 math coprocessor at 10Mhz
- Comes with CONCURRENT-DOS, the multitask-

ing operating system that is MS-DOS compatible

- Will directly boot PC-DOS**
- One year warranty on parts and labor

**PC-DOS may be booted and used to run programs that can be configured to run on terminals or some programs that restrict their operation to ROM BIOS calls for console I/O. Programs that directly access the video RAM will only operate satisfactorily when used in conjunction with our PC compatible color graphics board, COLOR MAGIC.

THUNDER/PLUS is ideally suited for any high performance 16 bit application, including but not limited to: high performance CAD systems, Scientific processing, data base applications, and process control applications. The 8087 coprocessor allows use of THUNDER/PLUS in applications not able to be run on previous 16 bit single board computers.

PRICE:

512K version.....	\$1195.00
1 Megabyte version	\$1395.00
8087 option.....	CALL

IBM COMPATIBLE S100 BUS COLOR GRAPHICS

LOMAS DATA PRODUCTS presents COLOR MAGIC, the most complete compatibility solution for S100 bus computer products. COLOR MAGIC includes three major hardware subfunctions which allow it to emulate the IBM-PC: An entirely compatible video function, an IBM-PC keyboard interface and an IBM-PC compatible timer interface including IBM-PC sound compatibility. HOW COMPATIBLE IS IT? Currently we are running MICROSOFT's FLIGHT SIMULATOR recognized as one of the severest tests of compatibility. We can also directly boot PC-DOS for the IBM-PC with no alterations. Other programs which have been tested and function without problems are: LOTUS 1-2-3, DBASE III, WORDSTAR and VOLKSWRITER.

COLOR MAGIC (16K byte version) . . \$595.00

■ THUNDER 186 single board computer provides a high performance 16 bit computer all on one board. It is an ideal companion to the COLOR MAGIC to provide a low cost, high performance IBM-PC compatible system. The 8Mhz 80186 offers 10Mhz 8086 performance. THUNDER 186 provides all the components necessary to form a complete system including: 256K bytes of no wait-state RAM, 2 serial ports, a parallel printer port, high performance floppy disk controller controls both 5 1/4" and 8" drives simultaneously, full IEEE 696 (S100) bus for system expansion. The COLOR MAGIC

and THUNDER 186 combine to provide performance you won't find in other compatibles. The price includes the powerful Concurrent DOS operating system. THUNDER 186 . . \$1195.00

■ NV-DISK is a solid state memory with software to emulate a disk drive under MS-DOS, Concurrent DOS, and CP/M-86. NV-DISK is entirely COMPUTRO software compatible allowing COMPUTRO users to take advantage of the lower cost and battery protection offered by NV-DISK. NV-DISK offers the advantage of high speed access and no moving parts. It can be battery protected so all data to stay even while powered down. The board is available in either 512K or 2 Megabyte configurations and multiple boards may be used to create disk drives with up to 16 Megabytes of storage.

NV-DISK 512K, \$595.00 2 MBYTE, \$995.00

■ MEGARAM is a high density, high performance dynamic RAM board with up to two Megabytes of storage. Megaram offers no wait state performance in 8086 systems with up to 10 Mhz processors at a fraction of the cost of comparable performance static RAM.

MEGARAM 1/4 MBYTE, \$475.00
1/2 MBYTE, \$595.00 1 MBYTE, \$795.00
2 MBYTE, \$1095.00

IBM-PC COMPATIBLE S100-BUS SYSTEM

LOMAS DATA PRODUCTS offers IBM-PC compatible systems with performance far ex-

ceeding that available from IBM. You can purchase systems offering performance of an eight Mhz 8086 or up the performance of an 8Mhz 80286. Each system is capable of supporting 8Mhz math coprocessors. Our 8Mhz 80286 system offers IBM-PC compatibility while offering up to 2 times the performance of the IBM-PC-AT. For applications where PC compatibility is desirable but higher performance is a benefit or requirement LDP offers the only viable solution.

WHO IS LOMAS DATA PRODUCTS . .

LOMAS DATA PRODUCTS has been shipping 16 bit microprocessor S100 bus products for five years. We have earned a strong reputation for reliability and performance over these five years. We were running MS-DOS (SCP-DOS/PC-DOS) before IBM knew the operating system existed. We offer a wide range of 16 bit operating systems including MS-DOS, CP/M-86, CONCURRENT CP/M-86 and CONCURRENT DOS. All our products are backed by a one year guarantee. We offer no 8 bit products and concentrate entirely on high performance 16 bit systems. If you are looking for the highest performance possible on the S100 bus, you can be sure LOMAS DATA PRODUCTS offers it.

The S100 Bus Experts.



Dealer inquiries invited.

MS-DOS, trademark of Microsoft
PC-DOS, trademark of IBM
Concurrent CP/M-86, concurrent DOS, trademark of Digital Research

LOMAS DATA PRODUCTS, INC.

182 CEDAR HILL STREET, MARLBORO, MASSACHUSETTS 01752 □ TELEPHONE: (617) 460-0333 □ TELEX: 4996272

Turbo Pascal Corner



by Stephen R. Davis

This column features tips and techniques for using Turbo Pascal productively on MS/PC-DOS and CP/M microcomputer systems. It discusses typical problems and their solutions. Reader suggestions, comments, and questions are encouraged. Address them to Turbo Pascal Corner, Route 5 Box 107K, Greenville, Tx 75401 or through MCI mail, 289-6124.

Beginning with this installment, I am assuming the reigns of the Turbo Pascal Corner from Dave Carroll. Long time readers of Micro/Systems Journal may remember some of my earlier articles, but I should nevertheless introduce myself. I am a full-time software engineer, part-time hacker and part-time freelancer. I have about 10 years experience with microprocessors ranging from 8080's to 68000's in assembler, Pascal and C.

I first became excited about Pascal as a computer language in 1978, when all higher level languages were limited to mainframes and BASIC was still king of the micros. I regained interest in 1983 when Turbo Pascal was introduced, not because of the language itself this time, but because of the incredible working environment. More working code can be generated in less time in Turbo Pascal than any other language environment. From time to time, I will be making appeals to BASIC holdouts to drop their security blankets and join us in exploring this powerful language tool. I will also be prodding Borland International into providing this same environment for other languages, such as C.

In future columns, I would like to examine various aspects of Turbo Pascal which may not always be clear. Borland did a reasonably good job in writing the user's manual for Turbo, but they could not include everything. Hopefully, I can explain and clarify aspects of the language not well covered in the manual. Example topics we will examine are: interrupt handlers (such as for communications, mouse drivers, etc.), manipulating screen memory directly, linked lists, the heap and Pascal's confusing file structure.

To be responsive to the reader's needs, I need your help. If there is an aspect of Turbo Pascal which you find particularly confusing, drop me a line and I will try and address it. Usually it helps to

Turbo Pascal Utilities In The Public Domain

explain concepts with examples, so I would also like to hear of projects you would like to see developed in future columns. I would like to hear from the CP/M TURBO users as well as the PC-DOSers (I have been watching for TURBO for the MAC and the Amiga; I will pass along anything I find out.)

Before we get started, however, you should be aware that there are several aids available to the PC-DOS Turbo Pascal programmer that you should not be without. (A list of commercial tools available was discussed in the January/February '86 column by Dave Carroll.) As mentioned above, the manual that came with your compiler disk is quite good, but there are some other quite useful manuals available on the market. The first comes from Borland itself, the *Turbo Pascal Tutor*. This book does not replace the original manual, rather it expands upon it by offering example programs of every possible Pascal construct. It is directed to the beginner up through the beginning expert and comes recommended for both groups. Although there is nothing in the Tutor which is not covered in the manual, seeing example programs worked out can save many a debug hour.

Another manual geared somewhat more towards the beginner is our very own Dave Carroll's book, *Programming in Turbo Pascal*. One of the nicest features of *Programming* is an included diskette containing, not only every example program from the book plus some extra utility programs, but also a version of the Turbo compiler to compile them with (this compiler is a limited feature version).

Besides books, software is also available in the public domain to help beginners and experts alike in their programming efforts. These tools are available free of charge (or nearly so) from your local Bulletin Board Systems and public

domain libraries (such as PC-Blue or the Turbo Users Group). Following are PC-DOS compatible highlights from the Turbo SIG (Special Interest Group) maintained on the CompuServe network by Borland International (many thanks to Larry Kraft).

THELP: Being bound, the Turbo manual will not lie open on your desk. TurboHELP addresses this shortcoming by placing most of the manual on-line. THELP must be installed before entering Pascal. Once installed, THELP may be brought up at anytime by entering Alt-R (some versions have been modified to use other key combinations such as Alt-H). THELP opens a window in the middle of the screen and displays a menu, much like the menu in the Pascal manual. By depressing the desired selection the operator can vector down to an example of almost any aspect of Turbo Pascal. Everything from edit commands to variable declarations to run-time error numbers can be found within THELP's menus.

THELP is not a replacement for the manual. It does not explain things to any level of detail; however, it is a tremendous supplement to the manual as it gives sufficient examples to jog the memory on most aspects of the language. It is interesting to note that THELP is written entirely in Turbo Pascal and that the source is also in the public domain. We will probably cover this topic again in a future column. (Dave Carroll's book includes an expanded version of this utility.)

TDEBUG: One of the most important tools in the hacker's toolkit is a decent debugger. Borland did not include a debug facility with Turbo, reasoning that since the compilations are so incredibly swift, it is sufficient to use the "Insert a WriteLn and Try Again" mode of debugging. Some problems just cannot be readily isolated this way. One of the most surprising finds in the public domain is TDEBUG (sometimes seen as DBGxx, where xx is a version number). TDEBUG is a simply incredible shell program which loads over TURBO.COM (or TURBO-87.COM). Once loaded, TURBO performs exactly as before except for the R(un) command, which now allows TDEBUG to take control, bringing up a debug screen.

TDEBUG divides the screen into two parts, with the source code listed above.

The currently executing line appears in high intensity. TDEBUG allows breakpoints to be set on source level statements and allows access to all currently defined variables. TDEBUG can also be run under an assembler debugger to allow breakpoints at the assembler level, if desired. TDEBUG only runs on IBM work-alikes and is specific to a particular version of TURBO 3.0 (that is, you must find the version of TDEBUG for your version of TURBO), but if you are serious about TURBO Pascal, this tool is simply too powerful to do without.

BEGEND: One of the most common errors in TURBO programs is mismatching BEGIN and END statements. By highlighting the beginning and ending of blocks clearly on the listings, BEGEND makes

Mail

continued from page 12

cause more damage than the transient one finds on the line. Considering true vs. switchover type, 8 mu transfer times appear to be too slow, but companies claiming 0.5 nano to 2 mu second transfer times are frequently blowing hot air.

I need a reliable power conditioner, and would like to have backup power as well. I do not want to pay \$2000 - \$4000 for ten minutes of reliable service from GPS or Topaz. (Why are the damned things so expensive? Is the price truly justified? Am I just *belly-aching*?).

Thanks,
Carl Voelz
Madison WI

We would be interested in publishing an article such as the one Carl describes. Anyone interested? Give us a call or drop us a line.

PC CONSTRUCTION ARTICLES

Dear Mr. Libes:

Like you, I have been driven to the conclusion that an IBM-PC (or clone) is a must nowadays. I hope that in your magazine you are able to get hardware articles on the PC. There seems to be a wall, built by Microsoft and the OEM, between the user and the inside workings of hardware/MS-DOS. Everything's OK as long as you can find/afford to buy the board that does what you want or you don't try to run 2 boards that conflict with each other.

I do not know how you were able to get such good construction articles for S-100 systems, but I hope you are able to do the same as we move through the IBM-PC age.

Denny Pistole
Port Hueneme, Ca

We are with Denny's sentiments and would like very much to publish construction type articles based on the PC bus. Anyone interested in writing such articles should contact us. We are also interested in publishing articles related to the PC bus and hardware. [u]

finding this class of error very easy indeed. Although fairly simple in design this little number could end up saving you lots of heart burn.

INLINE: For those familiar with assembler language, INLINE generates include files containing INLINE statements from an assembler source program. Although not identical, the syntax of INLINE is similar to that of the Microsoft MASM assembler. Best of all, this utility could end up saving you the expense of MASM.

WIND2: Another class of work savers is the subroutine library. This library of windowing routines (not in the Microsoft WINDOWS sense, more on this in future columns) is truly excellent. These routines are well documented for inclusion in any program, providing multiple, overlaying windows. These windows are much more powerful than Borland's half-hearted implementation.

EXEC: Executing one program from your own is another common function that is none too easy for the beginner. This library of DOS interface routines provides all of the most common DOS interface functions.

Although not absolutely necessary, any or all of these tools will aid the Turbo Pascal programmer. You might begin now looking for some of these if you plan to join us as we venture into some of the deeper and previously more confusing aspects of the language in future columns.

Another utility we will be examining in future columns is ALICE, a TURBO Pascal compatible interpreter. Interpreters, like BASIC, generally have superior debug and edit facilities. ALICE also has a context sensitive editor (that is an editor which understands Pascal). The programmer can develop his code using ALICE's Pascal editor and check his algorithms using ALICE's interpreter. Once the program is running correctly, it can be compiled and executed on the TURBO compiler directly from the ALICE environment. The result just might be drastically decreased debug times. I will be checking into this in a future column.

As a special incentive to send me your suggestions, I will send a disk containing the public domain programming tools listed above to anyone who submits a problem or topic for consideration which is good enough to be included in this column. I remain the final judge of what is worthy and what isn't, and mine is the final decision, but what do you have to lose? You just might get your problem solved and free software to boot, literally. [u]

Stephen Randy Davis is a senior systems programmer at a defense contractor in Greenville TX where he programs various microprocessors. He is also working on his Masters in Physics.

Does YOUR disk Look Like

THIS?



Let

H.D. Tuneup
cleanup your act!

H.D. Tuneup tunes your disk file system, making it better than new. File loads speed along like new. Directory searches are faster than new.

H.D. Tuneup reallocates all of your files to be contiguous. Each file is placed in its proper location, adjacent to its neighbors in the directory.

Disk operations run faster when your files have been Tuned.

H.D. Tuneup relocates your sub-directories into one area of your disk, as close to the root as possible. All deleted entries are actually removed.

Your drive head moves less from directory to directory than when your disk was new. TREE/F will fly when listing all of your files.

H.D. Tuneup requires IBM PC/XT/AT compatibility, 196k, and DOS 2.x/3.x. Fixed disks up to 32mb may be tuned, along with any 5.25" diskette.

NOT COPY PROTECTED!

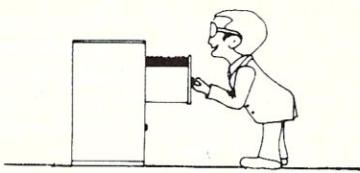
ONLY \$39.95 plus \$3.00 S/H

SofCap Inc.
P.O. Box 131
Cedar Knolls, N.J. 07927

Visa (201) 386-5876 M/C

N.J. residents add 6% sales tax.

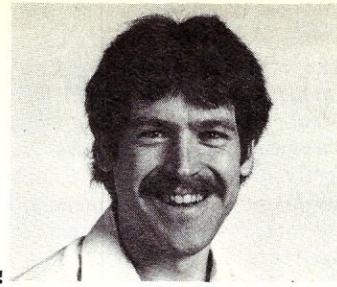
**For the best performance
from your disk
TUNE IT UP!!**



Visa, Master Card, Checks, Money Orders, or COD only.

The C Forum

by Don Libes



Garbage In

This column discusses the problems of reading input.

scanf

scanf does the opposite of printf. Well, sort of. scanf(format,arg1,arg2,...) reads "stdin" (by default, the keyboard) interpreting characters according to the "format". The results are stored in arg1, arg2, etc. sscanf() is just like scanf() except that it reads from the string given as its first argument rather than stdin. For example,

```
char name[20], first_word[20];
int age;
double iq;
char string[] = "Don 10 4.5 hello"
sscanf(string,"%s %d %f %s",name,&age,&iq,first_word);
```

assigns the value "Don" to name, 10 to age, 4.5 to iq and "hello" to first_word. Just like printf(), there are three different variations of scanf. The third one, besides scanf() and sscanf() is fscanf() which reads from a stream specified by its first argument. One of the big differences between scanf() and printf() is that arguments must be passed by address. Look back to the example and you will see "&age" rather than "age". If you think about it, you will realize that this is necessary. In order to have a subroutine store anything into "age" it must be passed by reference. Since the strings are passed by address automatically, they need no &.

The most common error using scanf is to forget the &. Typically, an uninitialized integer is used as a pointer, resulting in an address fault. In an unprotected operating system such as MS/DOS, a random area of memory will be destroyed.

```
int integer;
scanf("%d",integer); /* WRONG - need & */
```

scanf() is also confusing because it "only reads until the format is satisfied". Interactively, this can lead to problem programs such as:

```
for (i=0;i<MAX;i++) {
    printf("enter number %d: ",i);
    scanf("%d",&number[i]);
}
```

If the user types in a return by itself, scanf() will simply keep trying to read and never return, so the poor user never gets a response.

Changing the format to "%d\n" forces scanf to find the return, however this causes a new problem. scanf() treats returns equivalent to whitespace. Whitespace is defined as any number of spaces, tabs and returns (really newlines). Therefore, the user now has to type a return and the NEXT integer (and return) in order to terminate the whitespace! This is ridiculous.

One solution is to follow every scanf("%d") by a gets() to eat up the following newline. However a better solution, in general, is

to use fgets(), and THEN parse the string using sscanf() (after checking for an empty line). This way you get control back on an empty line and you always read whole lines.

Furthermore, this allows one to read other streams besides stdin, since fgets() takes a stream argument.

Even if you are just reading stdin, fgets() is preferred to gets(), since, without any checks, long input lines can overflow the user-supplied buffer. You should not assume that a newline terminates each buffer. In this way, you will correctly handle the last line in the file whether or not it is terminated with a newline.

Also, don't forget to check the return value from fgets(). If NULL is returned, the EOF was detected, and you should ignore anything in the buffer.

With the current stdio definition, there is one drawback to using this approach over the original scanf(). That is, if scanf() can not interpret stdin according to the format, it will leave the pointer where things stopped making sense. This is not possible with fgets() and sscanf(). However, the draft C standard defines a new %n format which places the number of characters scanned so far into the integer pointed to by the corresponding argument. Thus, in implementations conforming to X3J11/85-102, it will be possible to reparse data starting exactly from where the problem occurred.

Just for the record, none of these tips (or much of stdio) works on binary data. If you need to read a binary file, you should be using read() (or fread() for buffered input) explicitly.

HANDLING ARBITRARILY LONG LINES

But what if you don't know how long input lines can be? What if there are no limits on the length of the input, and yet, you would like to be able to store each line in a buffer?

That is a common enough problem, and here is a simple subroutine that solves it. big_fgets() is just like fgets() except that it only takes one argument, a pointer to a FILE. big_fgets() returns the address of malloc'd memory with the next line from the file. When you are done with it, don't forget to free() it! Like fgets(), big_fgets() returns NULL upon EOF, and preserves newlines.

```
#include <stdio.h>
#define GROW_BY 256
char *big_fgets(fp)
FILE *fp;
{
    int c;
    unsigned string_length = 0; /* including NULL */
    unsigned buffer_length;
    char *buffer, *malloc(), *realloc();
    if (!(buffer = malloc(buffer_length = GROW_BY)))
        return(NULL);
    while (EOF != (c = getc(fp))) {
        buffer[string_length++] = c;
        /* if no more space in buffer for more */
        /* chars increase the size of the buffer */
        if (string_length >= buffer_length) {
            buffer_length += GROW_BY;
            if (!(buffer = realloc(buffer,buffer_length)))
                return(NULL);
        }
        if (c == '\n') break;
    }
}
```

```

/* check for EOF */
if (string_length == 0) {
    free(buffer);
    return(NULL);
}
buffer[string_length++] = '\0';
/* Now that we know what size the string is, */
/* make do with a smaller buffer */
return(realloc(buffer,string_length));
}

```

The basic idea of big_fgets is that string space is allocated from memory in multiples of GROW_BY bytes. If more space is needed, the old buffer is realloc()ed by another GROW_BY bytes. Clearly, a careful choice of GROW_BY can be very important in terms of efficiency.

WAITING FOR INPUT THAT MAY NEVER COME

The opposite problem of handling lines that are impossibly long is handling lines that are impossibly short. In other words, what happens if you read() but nothing is waiting. Unless you make other arrangements, a read() will wait until some number of characters are available.

If you are reading from a terminal, read() will return upon encountering newlines no matter what length you specified. If you put the terminal into raw mode, read() will return as soon as any characters are available. However, if no characters have been entered, read() will wait forever.

There are no system-independent ways of solving this problem; however, by creating modular subroutines it is possible to be reasonably portable. In addressing this problem, the first issue is that there are several *styles* of checking for input. Few systems implement all of them in their full generality.

POLLED INPUT

Polling for input means that the program explicitly checks for input at certain times. It further implies that if no input is waiting, the program does something else rather than waiting for input to arrive.

BSD UNIX variants and many PC C compilers support an ioctl(FIONREAD) that allows one to check for the number of characters pending. In CP/M, there is a BIOS call (CONST) that tells you if characters are waiting. In MS/DOS, there is a BIOS call that plays the same role. Many compilers provide special functions such as kbhit() (Microsoft) that return true or false depending upon if a character is waiting. In each of these cases, the style of solution is to check whether characters are waiting before issuing the blocking read().

Alternatively, the behavior of read() may be changed so that it does not block if no characters are waiting. System III, System V, and UNIX 4.2 all support nonblocking reads. In the case that no characters are waiting, read returns -1, while errno is set to EWOULDBLOCK. Nonblocking I/O is set up with fcntl(). The parameters are typically system-dependent. Here is an example from 4.2 QDBSD:

```
fcntl(fd,F_SETFL,FNDELAY);
```

A more sophisticated technique supported directly by fewer systems is reading with timeouts. System III, System V and UNIX 4.2 support this kind of I/O. If the indicated time passes without the read() being satisfied, the read() returns. Reading with timeouts of less than one second can be done via ioctl() on System III and V and through select() on BSD 4.2 UNIX. Timeouts of greater than 1 second can be done portably between all UNIX systems with the following code:

```
{
/*1*/ signal(SIGALRM,alarm_handler);
/*2*/ alarm(5); /* timeout after 5 seconds */
/* start read() */
}
```

```

/*3*/ if(setjmp(context)) {
/*4*/     errno=EINTR; cc = -1;
} else {
/*5*/     cc = read(fd,buffer,count);
/*6*/     alarm(0); /* turn off alarm */
}

alarm_handler()
{
/*7*/     longjmp(context,1);
/* go back to setjmp and skip over read */
}

```

Briefly, this code does the following:

/*1*/ Registers a function to be called in the event of an *alarm clock* signal occurring.

/*2*/ alarm(5) sets up an interrupt 5 seconds from now. This signal will interrupt the read(), so that we can get control. In particular, the function registered at /*1*/ will be called when this happens.

/*3*/ setjmp() remembers where we are and what we are doing so that we can jump back to this point later. The first time we are here, setjmp() returns 0.

/*5*/ so that read() begins. If read() returns normally, we go to /*6*/ and turn off the alarm. However, if 5 seconds go by and read() has not completed, the alarm signal occurs and alarm_handler() is called.

/*7*/ If alarm_handler() is called, longjmp() resets the stack to the way it looked at /*3*/ and we find ourselves returning from setjmp(). What is returned is the second argument of longjmp(). Since the second argument of longjmp() is 1, we go to /*4*/ setting the appropriate variables to make it look as if read() has completed abnormally.

This is a good example of non-local goto's with setjmp/longjmp. If you have never used these functions before, let me point out the unusual side-effect of longjmp to *unwind* the stack. In this case, it caused us to *forget* that we were executing a read().

In a future column, I will talk more about setjmp/longjmp and show other useful applications of them.

INTERRUPT-DRIVEN INPUT

Often polling for input is inconvenient. Polling can also be inefficient. For example, if you are monitoring 100 devices that do I/O infrequently, but have to be serviced quickly when input occurs, you are going to waste a lot of time checking for input. The potential exists to miss servicing a device because you are busy checking others. A better solution in this case is to generate an interrupt when characters arrive.

In BSD UNIX, this is done by calling fcntl() which arms the device driver. When input arrives, a SIGIO signal is generated starting a user selected routine for catching such signals (set up with signal()). Here is an example from 4.2 BSD:

```
void (*sigio_handler)();
signal(SIGIO,sigio_handler);
fcntl(fd,F_SETFL,FASYNC);
/* direct signals to this process */
fcntl(fd,F_SETOWN,SIGIO);
```

In MS/DOS or CP/M there are no system independent primitives with which to request interrupt driven I/O. Under MS/DOS, it is possible to write a device driver or interrupt handler, although it will not necessarily be portable from one version of MS/DOS to another.

As an example, consider the IBM PC. Interrupt 9 of the BIOS is called upon every keystroke. Thus, it is possible for the user to install a keyboard interrupt handler by placing its address in the appropriate interrupt table entry. Typically, your interrupt handler

TASKVIEW™

WHY GIVE UP...

BATCH FILES,

I/O REDIRECTION,

SIDEKICK™,

DOS MENU PROGRAMS,

MOST OF YOUR RAM,

EXECUTION SPEED?

\$69⁹⁵ plus \$5.00 S&H

Washington residents add 7.9%
International orders add \$5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

Compatible, efficient DOS multi-tasking.

We designed Taskview with efficiency in mind. During normal operation, TASKVIEW hides behind DOS, providing you with control of up to 10 concurrent or non-concurrent programs. Just the touch of a key instantly switches a program to the foreground. Included desktop utilities let you cut and paste from program to program. Simple to use and reasonably priced, no well equipped PC user should be without it.

Requires: PC/AT/Jr compatible,
DOS 2.0-3.1, 256K RAM, 1
Floppy drive.

Taskview trademark of Sunnyhill Software
Sidekick registered trademark of Borland Int'l.

30-day money back guarantee

Dealer Inquiries Invited.

Sunny Hill
Software



13732 Midvale N. Ste. 206
Seattle, WA 98133
(206) 367-0650 M-F, 8-6 PDT

TURBO PROFESSIONAL™

SERVICE INTERRUPTS

No assembly required

RESIDENT PROGRAMS

Easy, pop-up routines

EXECUTIVE PROGRAMS

Run ANY DOS program

DISK SECTOR I/O

Lowest level access

FAST TEXT WINDOWS

Virtual windowing system

KEYBOARD MACROS

Simple, powerful

LOTS OF EXAMPLES

21+ full example programs

MUCH MORE...

Over 140 routines in all

\$69⁹⁵ plus \$5.00 S&H

Washington residents add 7.9%
International orders add \$5.00
VISA and Mastercard accepted.

To order Toll-Free
call 1-800-367-0651

"If you never thought Turbo Pascal was a systems programming language, you've never seen Turbo Professional."

Darryl Rubin
Computer Language

For programs that move with technology—Turbo Professional—a truly professional library of subroutines.

150 page reference manual.
Full source—many example programs.
No royalties charged for applications.

Requires IBM compatible,
DOS version 2.0 or greater,
Turbo Pascal 2.0 or greater.

Turbo Professional, trademark of Sunnyhill Software

Turbo Pascal, registered trademark of Borland International

Dealer Inquiries Invited.

Sunny Hill
Software



13732 Midvale N. Ste. 206
Seattle, WA 98133
(206) 367-0650 M-F, 8-6 PDT

will want to pass on characters it is not interested in handling by calling the original keystroke interrupt handler. The program should also clean up after itself at termination by restoring any interrupt vector entries that have changed.

Note that interrupt handlers must terminate with an IRET (return from interrupt) rather than a RET. Since the C compiler will generate RETs, if you wish to write the interrupt handler in C, you must code a small piece of assembler to do an IRET. Of course, you can't call IT because this will generate a return, also!

One technique (suggested by Phil Nanzetta) is as follows: Write an assembler routine c_interrupt() which decides whether to jump to the old interrupt handler, or the new C interrupt handler. If the C function is invoked, it returns to c_interrupt() when it completes. c_interrupt() then concludes with an IRET.

In a future column, I will go into this last topic in detail and present some working examples.

CONCLUSION

Rather than providing the volumes of information that this material deserves, I have been intentionally sketchy in the attempts to touch all bases. I have discussed all of the portable techniques for getting input, and I have briefly mentioned all of the typical nonportable tricks for avoiding nonblocking input from just about every common operating system. With what I have presented here, you should be able to get input in just about every way imaginable.

I encourage readers to write to me about topics or problems that you want to know about. I want this column to be reader driven. Write to me care of M/SJ, Box 1192, Mountainside, NJ 07092. [u]

Don Libes is a computer scientist working in the Washington, DC area. He works on artificial intelligence in robot control systems. He is also the son of Lennie and Sol Libes.

PORT-A-SOFT

Aardvark to Zorba We Can Handle It



**TRANSFER
PROGRAMS, DATA AND
OTHER FILES BETWEEN
OVER 400 COMPUTERS!**

13 OPERATING SYSTEMS

WORD PROCESSORS TOO!!

Prices From \$7.00 Per Disk
Software and Hardware So You Can Do Your Own

Call or write today for your
FREE CATALOG

P.O. BOX 1685 555 S. STATE, STE. 12
OREM, UT 84058 (801) 226-6704

Quality S-100 Products

Now Available! FULCRUM'S New...

256K STATIC RAM



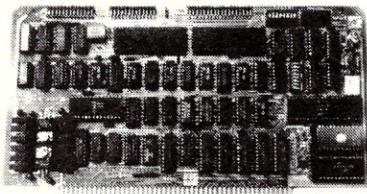
8/16 Bit Transfers
24 Bit Extended Addressing
Battery Back-Up

\$495

Designed For Speed & Flexibility.

MPUZ CPU

This NEW MPUZ CPU utilizes the Z-80 8MHz uPAs a basis for its 8MHz CPU for S-100 systems, and has been carefully designed to meet the requirements of the IEEE - 696 standard. The quality and performance this CPU provides is rarely found in S-100 products, and you can see why... **only \$299**



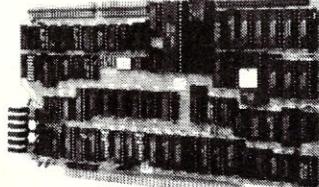
- 4 or 8MHz clock rate
- Two RS-232 serial ports
- Centronics printer ports
- Real time clock with battery back-up
- Vectored interrupts to any block location in memory
- Programmable timer
- ROM monitor

- Power on Jump
- On board wait states
- 2K of RAM space
- 24-bit extended addressing
- Latched Status
- Front panel compatibility
- MPM support

Best Value In Disk Controllers.

OMNIDISK

Now the FULCRUM OMNIDISK offers S-100 systems users a unique marriage of component compatibility and technological innovation. These together produce features not found in any conventional disk controllers made today. See for yourself what tomorrow looks like... **only \$299**



- Simultaneous support of both 5 1/4" and 8" floppy disks and hard disks
- Complete 24 bit DMA
- Power on boot for 5 1/4" and 8" floppy and hard disks
- Power on boot PROM
- On board de-blocking to save RAM space over BIOS
- Interfaces with the WD 1001[®]hard disk controller
- Supports 13 devices simultaneously
- Full track buffer allows controller to recall entire track
- DMA'S at 10 MHz
- Supports MS DOS
- 10K on board buffer saves two K of TPA

So before you buy another S-100 component, call or write for our **FREE** catalog. And see how your system can benefit from the FULCRUM difference.

*CP/M *2.2 configured for OMNIDISK \$60. *Trade mark of Digital Research. FREE U.P.S. ground shipping on prepaid orders. Shipping is added to VISA, M/C, and C.O.D. orders. CA residents, please add sales tax.

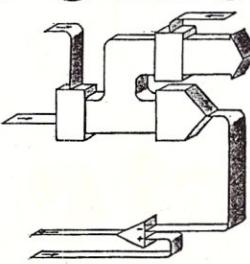


FULCRUM
COMPUTER PRODUCTS

459 Allan Court, Healdsburg, CA 95448.

Also in FULCRUM'S Family: OMNIRAM 64K memory board, Serial I/O 2-2 & Video I/O Interface boards, Relay board, I-8080, 8015 and 8035 main frames with 21 slot mother boards, CPA front panel and A/D board.

Control Systems Made Simple



by Lynwood H. Wilson

Computers are information handling devices. Typically, we feed them information. They process it according to some plan we have suggested to them. Then, we act upon the results. This is a good thing as it obviously benefits society and is one of the most complex and entertaining games yet invented. We can set up our own universes, our own rules and constraints, our own goals. We have the freedom to create anything we like. There need not be any particular correspondence with reality.

However, there is another mode. A computer can handle its own relationship with the real world, gather its own data, process it, make its own decisions, and produce whatever effect it has been told to consider desirable. We call such a system an intelligent controller. These systems vary tremendously in complexity. I will discuss a few of the simpler ways in which a computer can control a real world quantity. A temperature controller is considered here, but the principles will apply to almost any kind of control system.

Control systems are easy. You may not believe me, particularly if you have been corrupted by one of those enormously thick books on control system theory full of the most confusing kinds of mathematics, but it is true nevertheless. Laplace transforms indeed.

Don't misunderstand me. I approve of mathematics. I even like mathematics. I do not approve, however, of the philosophy that says "you can't do that unless you have these tools," or "without the proper background it would be foolish to attempt that." What is worse, those people admit in print that after you do your homework and figure everything out you will still have to make a few adjustments at the end. I propose that it is possible to start at the end and perhaps make one or two extra adjustments before you are through.

BANG-BANG CONTROLLER

In the simple temperature control system, the temperature is regulated by a heater which turns on or off according to whether the sensed temperature is below or above a set point. This process is similar to the one that operates the thermostat in one's home and is called an on-off or *bang-bang* controller. It isn't very accurate, but neither is your ability to sense temperature so it all works out well. The lesson here is not to provide more accuracy than the situation demands. Actually, a bang-bang temperature control can be quite accurate if the system lag and thermal inertia are small. (Lag is the time it takes before a change in heat input to the system shows up as a change in measured temperature and thermal inertia is the amount of change in heat input it takes to produce a particular change in temperature.)

My two cubic foot test chamber is a cardboard box nested in another cardboard box with two inches of foam packing separating the two boxes. Temperature is measured by an Omega model 199 digital thermometer with a resolution of 0.1°C. Temperature data is transported by a parallel BCD output to a pair of parallel computer ports. This thermometer can use any of a number of platinum resistance type sensors. The one I used here is designed for

harsh environments with its sensing element encased in a massive stainless steel tube, and so it reacts slowly to temperature change. If I hold the probe tightly in my hand for 30 seconds, the indicated temperature will not start to rise for several seconds. After letting go, the temperature will continue to rise for several minutes, and will take about 7 minutes to return to the original reading. The heater used dissipates 100 watts and is controlled by a solid-state relay controlled by bit 0 of port 0 on my computer.

Program I is a simple BASIC (Microsoft Basic-80 ver5.22) program which enables this system to act as a simple on-off controller. The function FNtemp defined in line 20 gets the temperature. It sends the BCD data from the thermometer through ports 0 and 1. The low nybble (four bits) of port 0 represents tenths of a degree; the high nybble is degrees. The low nybble of port 1 is tens of degrees, and the high nybble of port 1 (which we discard) contains a bit representing 100 degrees and a strobe.

Program I Bang-Bang Temperature Control

```
10 setpoint=40
20 DEF FNtemp=(10*(INP(1) AND 15))+((INP(0)
     AND 240)\16+((INP(0) AND 15)/10)
30 'function reads temperature from thermometer in
40 'BCD (ports 0 & 1) & translates it into decimal
50 IF FNtemp < setpoint THEN OUT 0,1 ELSE OUT 0,0
60 ' port 0 bit 0 controls heater
70 GOTO 50
```

First, the tens digit is recovered with (INP(1) AND 15), which inputs the data at the port and discards the high nybble by ANDing with 15 (00001111₂). Then, the digit is multiplied by 10. (It is the tens digit, remember?) Next, the ones digit is recovered with (INP(0) AND 240), 240 (11110000₂) discards the low nybble. Dividing the result by 16, moves it to the low position. (Note that in this dialect of BASIC the backslash represents integer division and the slash represents floating point division.) Finally, the tenths digit is found with (IMP(0) AND 15), discarding the high nybble. The result is divided by 10 and added to the other two numbers. The rest of the program is straightforward, merely turning the heat on if the temperature is lower than the setpoint, and off otherwise.

Because of the sensor temperature lag, chamber temperature can be several degrees ahead of the thermometer reading when it is changing rapidly. For instance, when the indicated temperature is 39.9°C and rising, the true temperature is 42°C and rising at about the same rate as the indicated temperature, and the heater is on. (Part of the reason for this lag is the extremely poor heat transfer between the probe and the still air in the chamber. System performance could be improved greatly by adding a small fan to stir the air. We did not do this since wanted to make our test conditions as difficult as any situation likely to be encountered.)

This situation is clearly not conducive to stability causing the thermometer to produce an oscillation between 39.4 and 41.0°C. The actual air temperature swings a good deal further and is one of the major causes of instability. Though it cannot be eliminated, we can program the control force to lead the signal so that this lag no

longer bothers us. (The high and low temperatures are not symmetrical about the setpoint because the air heats up faster than it can cool. Ideally these two effects should be approximately equal.)

If I replace the temperature sensor with another that has its element deposited on a tiny ceramic chip exposed directly to the air, the thermometer will react much faster to changes in the test chamber and so Program I will work better, holding an indicated temperature of 40.0 with an occasional flash of 39.9°C.

See how easy it is? We have only started and already we can control the temperature as closely as we can read it. However, this is a special case where control is a simple task since the test chamber has very little thermal inertia and our measurement system has such a small lag time. In general, by keeping thermal inertia and the measurement lag to a minimum, control will be easiest. It should also be noted that the faster responding sensor probably gets considerable infra-red radiation from the heater, which helps to increase its response speed, but does not help it follow air temperature. This effect is a kind of negative lag, as the sensor responds to the heat before it does to the air. This can be quite beneficial to stability as long as we are aware of the limitations. In an extreme case, this kind of situation can yield a correct and stable reading from the sensor while the actual quantity is stable at quite a different point. We must be sure that what we are measuring is really what we want to control.

PROPORTIONAL CONTROLLER

In order to more nearly approximate real world conditions, we will return to the stainless temperature probe. Its lag will give us a chance to find out whether we can control the temperature of a less than ideal chamber. As the first test showed, the bang-bang controller will not stabilize in this situation. Since the lag of the sensor is directly related to the rate of temperature change, we could improve stability by reducing the heater rate, but that would increase the time to get to the setpoint. We will try reducing the heat in proportion to the deviation from the setpoint, the error, which should improve stability without reducing rate of control at large errors. This approach is called a *proportional controller*.

Program II Proportional Temperature Control

```

10 setpoint=40
20 DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND
240)\16)+((INP(0) AND 15)/10)
30 heat=10*(setpoint-FNtemp)
40 FOR I=1 to 100
50 IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
60 heat=heat-1
70 NEXT I
80 GOTO 30

```

This program is designed to turn the heater off if the temperature is above the setpoint, and turn it on if the temperature is more than 10°C below the setpoint. Between these two points, the heater turns on and off with a duty cycle (percentage of time turned on) that is proportional to the error, i.e. deviation from the setpoint.

In Program II the variable *heat* increases as the temperature falls farther below the setpoint, ranging from 0 at the setpoint to 100 at ten degrees below the setpoint. This represents the desired percentage of heater on-time. The heater is never turned on if *heat* is 0 or negative, which is to say if the temperature is greater than or the same as the setpoint. Otherwise, the heater stays on until the variable *heat* is decremented down to zero, taking more and more of the 100 iterations of the FOR loop as the temperature decreases. When the temperature is 10 degrees cooler than the setpoint, the heater stays on constantly. In the case of the program above with a setpoint of 40 degrees, if the temperature is 42, *heat* will be -20, i.e. $(10 * (setpoint - FNtemp))$. Therefore, the heater won't turn on since *heat* is not greater than zero at the first pass of the FOR loop. As the FOR loop goes through its 100 iterations, *heat* will be decremented each time, and the heater stays off. If the temperature

drops to 37°C, the value of *heat* will be 30 during the first iteration of the FOR loop and the heater will be turned on at the first pass. Heat becomes 29. The loop continues leaving the heater on and decrementing heat until it is zero which will occur during the thirtieth pass. On the thirty-first pass, heat will be zero and the heater will be turned off, having been on for thirty iterations. The loop continues with the heater off for 70 more iterations before it is done. If the temperature is less than 30 degrees, heat will be positive for all 100 iterations of the FOR loop, and so the heater will always be on. With a proportional controller, the heater remains on for a percentage of the time directly related to the heat value.

The range over which the heater is proportionally controlled is called the *proportional bandwidth (pb)*. The following small modification to the previous program allows one to specify the proportional bandwidth directly in degrees.

Program III Proportional Bandwidth Control

```

10 setpoint=40
20 pb=10
30 ' proportional bandwidth
40 DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND
240)\16)+((INP(0) AND 15)/10)
50 heat=(100/pb)*(setpoint-FNtemp)
60 FOR I=1 to 100
70 IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
80 heat=heat-1
90 NEXT I
100 GOTO 50

```

The change is in line 50 where 10 is replaced by 100/pb. The 10 was there to give a reasonable proportional bandwidth for the last test. Now the proportional bandwidth is the number entered for pb. Remember that when *heat* ≤ 0 , the heater stays off; when *heat* ≥ 100 , the heater stays on. Thus, if pb = 5, *heat* will be 0 when the temperature equals the setpoint, and 100 when the temperature is 5 degrees below the setpoint. Thus, the proportional bandwidth is 5 degrees.

If you have a computer that can control something, try changing the proportional bandwidth. It is important to try these controllers as you go along because in the end, what we are going to substitute for all that odious mathematics, is your experience and intuition. Now is the time to get some.

OFFSET

My system finally stabilized at 37.5°C after quite a time of wandering aimlessly. It then started climbing slowly as the ambient temperature began to rise. Program III is closer to stable than the bang-bang type, but still isn't stable enough; it doesn't even oscillate around the setpoint when it oscillates. Not good. We must find a way to get the temperature to the setpoint. Program III puts out no heat at the setpoint. Obviously it cannot maintain the temperature at that value. We must add an offset to move the temperature up to the setpoint. It can be done like this:

Program IV Proportional Bandwidth With Offset

```

10 setpoint=40
20 pb=10
30 offset=50
40 DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND
240)\16)+((INP(0) AND 15)/10)
50 heat=(100/pb)*(setpoint-FNtemp)+offset
60 FOR I=1 to 100
70 IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
80 heat=heat-1
90 NEXT I
100 GOTO 70

```

In line 30 *offset* is assigned the value 50. In line 50 it is added to the equation that gives a value to the variable, *heat*, effectively increasing its value by 50. This results in a 50% duty cycle for the heater at the setpoint, when *setpoint - FNtemp* = 0, 100% duty cycle at 5°C below setpoint, and 0% duty cycle at 5°C above setpoint. The bandwidth still equals pb, although now it has been moved to center about the setpoint.

Though Program IV is an improvement, it is still not terribly stable, but at least it hovers near the setpoint. This proportional controller does not hold the temperature at exactly the setpoint, however. The amount of heat supplied to produce an error of 0 is not exactly right to maintain the temperature. Even if we had been lucky and guessed exactly the right offset, it would only have worked until something such as the ambient temperature changed causing the system to drift away from the setpoint.

RESET

We need some sort of self-aligning force. We will take a portion of the error (deviation from the setpoint) and add it to the offset at each pass through the program. (We must ensure that the sign is correct: error above the setpoint is negative, below is positive. The error used (line 70) is $(setpoint - FNtemp)$, which is correct.) This won't be much at first, but as time goes on if the error persists, the effect of this offset will become greater until it finally reduces the error to zero. This is traditionally called *reset* and can be accomplished as follows:

Program V And Reset

```

10  setpoint=40
20  pb=2
30  offset=30
40  reset=5
50  DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND
     240)\16)+((INP(0)AND 15)/10)
60  heat=(100/pb)*(setpoint-FNtemp)+offset
70  offset=offset+((setpoint-FNtemp)/10)*reset
80  FOR I=1 to 100
90    IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
100   heat=heat-1
110   FOR T=1 to 0
120   NEXT T
130 NEXT I
140 GOTO 60

```

Lines 110 and 120 are merely a time delay. We will need it later and I put it in here so we can keep the same numbers from here on. I have attempted to keep the weight variables, such as *reset*, between 1 and 10 for our convenience. Of course, there is no assurance that they will be the same for other systems.

In line 40, *reset* is given a value. In line 70, a small portion of the error is added to the offset each cycle. The constant, *reset*, controls the rate at which the offset is changed in response to a persistent error. Too much *reset* will cause instability, too little will increase the time required to reach the setpoint or to respond to a change. Once again, experiment with this new control algorithm.

My system isn't very steady with Program V, but it does stay closer to the setpoint than previous tries. At least the system seems to know where the setpoint should be and hovers around it. However, if the system is initially very far from the setpoint, it will take quite a while to stabilize, swinging widely to either side. Even when the system appears to have settled down, movement will still occur for no apparent reason.

RATE

The instability above results largely from *reset*, which is a kind of positive feedback leading to oscillation. In order to damp the oscillation, negative feedback must be added. This will take the form of a function that opposes change, a force that resists in direct proportion to the rate of change. It is called *rate* or occasionally referred to as *approach*. It is no more difficult to implement than the others.

Program VI With Rate

```

10  setpoint=40
20  pb=2
30  offset=30
40  reset=5
50  ratewt=4
60  DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND 240)\16)+
     ((INP(0)AND 15)/10)

```

```

70  oldtemp=FNtemp
80  ratecorrection=(oldtemp-FNtemp)*100*ratewt
90  oldtemp=FNtemp
100 heat=(100/pb)*(setpoint-FNtemp)+offset+ratecorrection
110 offset=offset+((setpoint-FNtemp)/1000)*reset
120 FOR I=1 to 100
130 IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
140 heat=heat-1
150 for T=1 TO 100
160 NEXT T
170 REM 150-160 are just a time delay.
180 NEXT I
190 GOTO 80

```

Ratecorrection is a measure of the rate of change with the sign reversed, so that it opposes the change. Note that if we did not initialize *oldtemp* in line 80, we would have a nasty first-cycle glitch as the program reacted to the sudden jump in temperature from 0 (uninitialized variable) to whatever *FNtemp* gets read the first time around. Note also that the time delay controls the maximum effect of rate. Rate can only react to the temperature change since the last cycle, and can only effect the heat output during the current cycle. By lengthening the cycle, rate is given a longer time to hold the heat on or off in response to the rate of change. Once the proper interval is determined, it can be fixed and the constant, *rate*, used to vary the weight of the rate function.

Reset must also be corrected for the change in cycle time. When cycle time is increased, reset happens less often, and therefore has less effect. The other factors are largely insensitive to cycle length, unless it becomes so long that the thermometer can follow it.

ANTI-WINDUP

The system seems a great deal more stable now, but when approaching the setpoint from a great distance, reset can build up such an offset that the temperature overshoots the setpoint badly before settling down. In many cases, this can be an asset since the artifact being heated and cooled may lag the air temperature and come to stability at the new temp more quickly after a substantial overshoot. The lesson here is to be sure you know what you want, whether it be a number on a thermometer or the core temperature of a block of metal. If overshoot is not desirable, it can be reduced by increasing the rate factor, but that will slow all temperature change. Better to add another refinement to our program. This one is called *anti-windup* and it has the effect of disabling the reset function when the temperature is outside the proportional band. In other words, when the temperature is so high that the heater is not turning on, or so low that it is not turning off, the offset does not change. This makes sense since nothing *reset* does can affect the speed at which the temperature approaches the setpoint under these conditions.

In systems relatively free from changes in ambient conditions, *reset* can be made very small. This is good for stability. An accurate temperature chamber is one which is built inside a well insulated chamber so that the inner chamber is not influenced by changes in ambient temperature.

Program VII Featuring Anti-Windup

```

10  setpoint=40
20  pb=2
30  offset=30
40  reset=5
50  ratewt=4
60  DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND 240)\16)+
     ((INP(0)AND 15)/10)
70  oldtemp=FNtemp
80  ratecorrection=(oldtemp-FNtemp)*100*ratewt
90  oldtemp=FNtemp
100 heat=(100/pb)*(setpoint-FNtemp)+offset+ratecorrection
110 IF (heat < 100 AND heat > 0)
     THEN offset=offset+((setpoint-FNtemp)/10)*reset
120 FOR I=1 to 100
130 IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
140 heat=heat-1

```

```

150 FOR T=1 TO 100
160 NEXT T
170 NEXT I
180 GOTO 80

```

In lines 120 and 130, the reset correction is only added to the offset if heat is between 0 and 100. This prevents large excursions of offset, while the error is great.

Once again play with the control numbers and get a feel for the effect of each of them. This program, with the proper constants, should control the temperature in most situations nearly as closely as it can be read. It is extremely useful to have a chart recorder or a program that can plot temperature as the constants are changed. Knowing where you have been is essential to knowing where you are going.

This program works well, but on a long approach to the setpoint, it still builds up sufficient offset to cause it to overshoot by half a degree or more. This may well be good enough, but in case it isn't, I know one more trick. The offset should not be changed during the approach. To prevent this, we need only add a statement that prevents the operation of the reset feature when the temperature is changing. In this manner, the offset will remain constant until the temperature stabilizes. Reset will then begin to bring the temperature to the exact setpoint. This can be accomplished in the following fashion:

Program VIII Featuring Still More Anti-Windup

```

10 setpoint=40
20 pb=2
30 offset=30
40 reset=5
50 ratewt=4
60 DEF FNtemp=(10*(INP(1)AND 15))+((INP(0)AND 240)\16)+((IN(0)AND 15)/10)
70 oldtemp=FNtemp
80 ratecorrection=(oldtemp-FNtemp)*100*ratewt
90 oldtemp=FNtemp
100 heat=(100/pb)*(setpoint-FNtemp)+offset+ratecorrection
110 IF (heat < 100 AND heat > 0) AND ratecorrection = 0
    THEN offset=offset+(setpoint-FNtemp)*reset
120 FOR I=1 to 100
130   IF heat > 0 THEN OUT 0,1 ELSE OUT 0,0
140   heat=heat-1
150   FOR T=1 TO 100
160   NEXT T
170 NEXT I
180 GOTO 80

```

The change was made in line 110 where reset only works when the temperature is within the proportional band and not changing, i.e. when ratecorrection = 0. I also deleted the division by ten in the calculation of the offset in line 110 since the restrictions placed on reset allow it to be more powerful without causing trouble.

This program does all one could want with the system I have described. It can come from ten degrees below the setpoint at full heat, shut off about two degrees away from setpoint, then go directly to the setpoint with perhaps one tenth of a degree of overshoot. But to do this each of the weight numbers, pb, offset, reset, and ratewt, must be adjusted to the correct values for the system involved.

TUNING

Rate is one of the reasons that it is desirable to size the heater so that at the setpoint, the heater is on about half of the time, i.e. a 50% duty cycle. Then, when approaching the setpoint from either direction, rate can apply the same control authority to slow the approach thereby preventing overshoot and oscillation. In the case of controlling to a temperature just above ambient, there is little control authority in the cool direction and much in the heat direction, thus maintaining a constant temperature becomes quite a difficult problem. If the temperature drifts higher, the program calls for less and less heat, but to no avail as the heat is already off. This kind of instability can be overcome to some extent in the programming, but the difference in weight between cooling and heat-

POWER USERS - PROGRAMMERS

Now, in one integrated utility, a super powerful DOS Shell and the best disk management tool available. **XPIP** provides powerful replacements and extensions for most DOS commands plus easy command recall and editing. Execute any DOS command or application program from within the **XPIP** Shell. Traverse the directory tree with single keystrokes. Create and access hidden files and directories. Easily edit any file on the disk, even .EXE files, in either HEX or ASCII. Complete online help plus either menu or command line operation. Too many features to describe in any size ad. Now includes a powerful file sort utility. Not copy protected. CP/M version also available. All at a "public domain" price of \$29.95 + \$3.00 s&h. You'll wonder how you got along without it. Order your copy of **XPIP** today.

System Facilities, Inc.

P.O. Box 7079 (M)	P.O. Box 5664 (M)
Charlottesville, VA 22906	Lacey, WA 98503
(804) 977-5245	(206) 456-0628

MasterCard and Visa Accepted	
VA sls tx: \$1.20	WA sls tx: \$2.25

ing requires different constants for rate and reset depending on whether the temperature is above or below the setpoint. This can be done, but by far the easiest and most straightforward solution is to add a cooler which will operate at a constant rate such that the heater (under control from the computer) runs at a 50% duty cycle to maintain the setpoint. It is much easier to control a system balanced between two equally strong opposing forces. Increasing the control authority in this fashion is equivalent to reducing the thermal inertia of the system.

Now that the system is built and programmed, the constants for each portion of the program must be determined. There are, as I said earlier, ways of doing this mathematically which work pretty well. If your work demands great precision, I recommend the mathematical approach. Otherwise, let's get on with tuning the controller.

First and easiest to determine is *offset* which should be set to the number that will hold the setpoint temperature. Once the rest of the constants have been set close enough to allow the system to stabilize, the value of offset can be read from the variable as follows: ^C to stop the program, then "PRINT offset" will give the current value of offset.

Next, the *proportional bandwidth (pb)* is tuned. As the pb is reduced, the system becomes more resistant to displacement caused by changes of any kind. For example, the temperature will not move as far from the setpoint as a result of a change in heat flow into the chamber which may be due to a change in the ambient temperature or a change in the heat produced in the chamber (you did intend to have something in there, didn't you?). Since reducing the pb reduces the distance the temperature will swing, the effect of reset is reduced which gets stronger as the temperature gets further from the setpoint. This improves stability and accuracy, and is an unmitigated *good thing* unless the temperature swings out of the proportional band. When the temperature gets out of the



POOR MAN'S NETWORK

Now you can implement networking on your own CP/M computers and share resources whenever you want. Each user can access files and printers on the other computer directly, without expensive hardware or switches, and without a communications program. Share floppies, RAM-disk, hard disks, and printers between two users. Works with most standard CP/M programs, like Wordstar, PIP, dBase II, etc.

Poor Man's Network works best with bidirectional parallel ports, or standard serial ports at 9600 baud or greater. Present version supports two computers only, and requires CP/M 2.2 on each computer. Uses only 6K of memory. Does not require a hard disk; runs on 8080, 8085, Z-80, HD-64180.

Poor Man's Network comes ready to run on BigBoard I and II, Xerox 820, NorthStar Horizon, NorthStar Advantage, Televideo TPC I, Kaypro (not 2000), Apple II (Super Serial required), S-100 with CompuPro Interfacer 4. Can be installed on other computers by altering one of the assembler overlay files provided. Each disk contains drivers for all the specified computers.

Best of all
is the price: only **\$69!**

Specify disk format: 8" SSSD, NorthStar, Kaypro, or Apple CP/M. Phone orders accepted; sorry, no CODs or Purchase Orders. Payment may be by certified cheque, money order, Visa, or Mastercard. Personal cheques from outside Canada require up to 6 weeks to clear. Canadian orders please pay in Canadian dollars; others in US dollars. Price includes diskette, manual and postage. Ontario residents please add provincial sales tax.

Note: If you use another operating system, such as CP/M 3 or PC-DOS, send us your name and address, but no money, and we will notify you when it is available for your system.

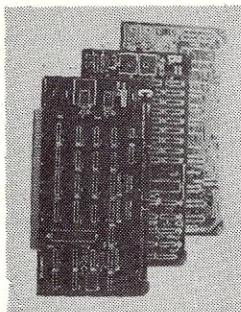
ANDERSON TECHNO-PRODUCTS INC.
947 Richmond Road, Dept S
Ottawa, Ontario K2B 6R1, Canada

Telephone **613-722-0690** for more information or to order.

CP/M is a trademark of Digital Research; Z-80 is a trademark of Zilog; Wordstar is a trademark of MicroPro; dBase II is a trademark of Ashton-Tate. Poor Man's Network is a trademark of Doug Anderson Techno-Products Inc.

**S100
BARE BOARDS**
ANY COMBINATION OF
3 BARE BOARDS

\$150



	B/B	KIT	ACT
256K Dynamic Memory 64K to 256K to 1 Megabyte		256K/510 64K/360	
Capacity	\$75		
FDC-1 Floppy Disk Controller 8" & 5 1/4" Drives	\$70	\$245	N/A
Z80 CPU 4 MHZ/2 MHZ Selectable	\$65	\$245	N/A
I.O. Board - 4 Serial Port Uses 8251 US ART	\$65	N/A	N/A
S100 Clock/Calendar with On Board Battery Backup	\$50	\$99	\$135
Source Code Monitor On Standard 8" Disk Used With CPM 2.2			\$25
Monitor In PROM - 2716			\$25
BUY ANY COMBINATION OF 4 BOARDS & RECEIVE FREE MONITOR PROM			

All Manuals Included - All Parts Available - CA Residents Add Sales Tax
\$4 Min. Shipping - Add \$1.65 COD
We Accept Credit Cards

COMPUTIME

8371 MUNSTER DRIVE, HUNTINGTON BEACH, CA 92646

(714) 536-5000

Business Hours 10-5 Monday through Thursday

pb, the system begins to resemble the bang-bang controller, and becomes less stable. Narrow the pb as much as possible and stay within it. However, in a system with a great deal of lag, too narrow a pb can be destabilizing. Try changing the parameters one at a time until you find out what works for your particular situation.

Reset is destabilizing. As it is increased, the tendency toward oscillation is also increased, but the time required to regain the setpoint after a change in conditions is decreased. Note this is quite different from the effect of pb on the magnitude of the error produced. To simplify, if the ambient temperature changes suddenly, the temperature in the chamber will also change. The smaller the pb, the smaller the magnitude of that change. The controller will immediately act to return the temperature to the setpoint through the reset feature. Therefore the time required to regain the setpoint is dependent upon the value of reset. However, this is more theoretical than practical, since under actual operating conditions other effects can swamp these to such an extent that they seem reversed. In any case, reducing reset always increases stability.

The last control variable is **rate**, which is primarily a stabilizing influence. It resists change preventing oscillation, but at the cost of slowing response to changes in operating conditions. In general, rate should be increased to the value at which oscillation ceases. Any increase in rate beyond this point, unnecessarily decreases response to change. The influence of rate is strongest near the setpoint, since at a greater distance (as determined by pb) the effect of rate is overcome by that of error (see Program VI).

Unfortunately, each of these parameters has an effect on the operation of the others which is complex and not at all obvious. Equally unfortunate, is that it takes a long time to see all the results of a change, particularly in the case of a system with a great deal of lag and inertia. With patience and a plot of past behavior, it will all become clear eventually. The second time is much easier.

I hope these ideas have opened up a whole new range of things for you to do with your computer, as they have for me. I am currently working on a program to operate a vacuum system and control pressure over a wide range, and am thinking about applying these principles to robotics.

In closing, I would like to mention two books that have been very valuable to me. For designing and building interfaces between your computer and the rest of the world *Interfacing to S-100/IEEE 696 Microcomputers* by Sol Libes and Mark Garetz is just the thing. It takes you through the theory in detail, and offers many examples which can be used with no modification. And for control system theory (and a good deal more), my favorite is *Real Time Programming* by Caxton C. Foster. This book is remarkable for covering so much material and yet being easy to read. (H)

Lynwood Wilson lives in a log cabin in the Rocky Mountain National Forest. The wood stove is not computer controlled, as he feels certain jobs are too good for computers. He is a founder and officer of Mandarin Systems Inc., in Research Triangle Park NC, developers of expert systems for almost any purpose. His hobbies include airplanes, motorcycles, science fiction, and not living on the east coast. He is building a robot.

**Advertise in
November/December
Issue**

**Reservation deadline:
September 23rd**

**For details contact:
Lennie Libes
(201) 522-9347**

COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

Competitive Edge brings you Lomas Data Products Inc.

S-100 CIRCUIT BOARDS

Thunder 186 , 512K, 2 Serial, 1 Parallel, Floppy Disk Controller, Clock, All on One Board with Concurrent DOS	\$895
8MHX Lightning 286 CPU	\$756
10MHZ Lightning 286 CPU	821
Control It All Floppy & HD Controller	487
Hazitall 2 Serial, 2 Par, Clock	260
256K Dram Megaram	358
1024K (1 Megabyte) Megaram	599
Ram 67 128K Static 100ns Chips	596
Octaport 8 Serial to 38.4K	320
NV Disk 512K, Memory Drive	\$371
Concurrent DOS Single User	280
MSDOS™ for 86, 186, or 286	200
CPM22EM 8080 Emulator	55
6MHZ Lightning 286 CPU	756
8 MHZ Lightning 8086	420
LDP Floppy Contr.	220
Color Magic Color Graphics Board	476
512K Dram Megaram	\$448
Thunder + 186 512K 10MHZ	995
Thunder + 186 1024K 10MHZ	1146
Octaport 4 Serial	200
NV Disk 2048K Wow!!	746
Concurrent DOS Multi-user	360
LDP-COM Modem Program	55
PC DOS Drivers	35

COMPETITIVE EDGE INTEGRATED SYSTEMS

THUNDER 186, 4 SLOT PC STYLE CABINET, 2-5" FLOPPYS, CDOS 512K	\$1595
THUNDER 186, 4 SLOT, 20 MB HARD DISK, 1-5" FLOPPY, CDOS 512K	2595
THUNDER 186, 4 SLOT, 20 MB HD, 1-5" FLOPPY 4 USER 1024K	3195
THUNDER 186, 4 SLOT, 2-5" FLOPPY, COLOR MAGIC, KEYBOARD 512K	2250
8MHZ 286, 1-5" FLOPPY, 1024K, 20MB HD, 10 SERIAL, 15 SLOT, 7-8 USERS	4695
8MHZ 286, 1-5" FLOPPY, 1024K, 40MB HD, 10 SERIAL, 15 SLOT, 7-8 USERS	5295
10MHZ 286, 1-5" FLOPPY, 1024K, STATIC, 40 MB HD, 10 SERIAL, 7-8 USERS	7550
10MHZ 286, 1-5" FLOPPY, 512K STATIC, 40 MB HD, 2 SERIAL & 80287	6495
8MHZ 8086/8087, 1-5" FLOPPY, 512K DRAM, 20 MB HD, 2 SERIAL CDOS or MSDOS	3695
10MHZ 286 1-5" FLOPPY 1024K 120MB HD 10 SERIAL SLOT 7-8 USERS CDOS	8495

LOMAS STATIC RAM **NEW** **256K TO 1024K** **PRICES START AT \$412**

FEATURING CompuPro® FROM VIASYN™

S-100 CompuPro® CIRCUIT BOARDS

Disk 1A™ Floppy Disk Controller	\$446	Network 100™ Network Board	\$371	Interfacer 4™ 3 Serial 1 Par	\$338
8 MHZ 286 CPU A&T 2 CYL.	671	80287 Option for 286 CPU	199	CPU Z™ 8MHZZ80	207
Ram 22™ 256K Static Ram A&T	446	Concurrent DOS™ 8-16™	750	MDRIVE-H 2048K, 2MB	821
Ram 23™ 64K Static Ram A&T	189	CP/M® 2.28" or 5"	189	Keytronics 5150 for PC Video	169
Interfacer 3™ 8 Serial Ports	446	Disk 3™ Hard Disk Controller	521	Passive Hub for Network 100	65
SPU Z™ 8MHZ Z80, 256K Multiuser	296	10MHZ 8088, 85/88 CPU A&T	265	HX-12 Hi-Res Color Monitor PC	459
MDRIVE-H™ 512K Memory Drive	371	Ram 23™ 128K Static Ram A&T	263	CP/M® 8-16™	268
PC Video Board for IBM® Compat	396	System Support 1™ A&T	338	SP186 512K Slave	495

HARD DISK SUB-SYSTEMS

40 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM \$1950
 80 MEGABYTE HARD DISK, CABINET, DISK 3, SUB-SYSTEM 2450
 ALLOY 17 MB TAPE BACKUP 1795
 40 MEGABYTE HARD DISK, CABINET, DISK 3, 5" FLOPPY 2150
 8" & 5" FLOPPYS, 40 MB HARD DISK SUB-SYSTEM 2650
 2-40 MB HARD DISK DRIVES IN CABINET, DISK 3, SUB-SYSTEM 3095
 SYSTEMS INTEGRATED BY COMPETITIVE EDGE WITH COMPUPRO® COMPONENTS

8MHZ 286 SYSTEM	\$5895	10MHZ 8088-85/88 SYSTEM	\$4495
8MHZ 286 & 512K STATIC RAM, 9 SERIAL PORTS ONE 5" FLOPPY & 40MB HARD DISK, DISK 1A & DISK 3 CONCURRENT DOS 8-16 MULTI-USER, 15 SLOT CABINET EM8080 SOFTWARE 8 BIT EMULATOR		85/88 CPU & 128K STATIC, 4 SERIAL & 1 PARALLEL SYSTEM SUPPORT & I/O 4, CU 5" FLOPPY & 40 MG CP/M-8-16 8 & 16 BIT, 15 SLOT CABINET SYSTEM UPGRADEABLE TO MULTI-USER & Z80 SLAVES	
8MHZ 286 SYSTEM	\$6295		

AS ABOVE BUT 80 MB HD INSTEAD OF 40 MB HD

BASE AT™ COMPATIBLE 8 MHZ, 512K, 1.2 MB FLOPPY

\$1295

FLOPPY & HARD DISK CONTROLLER, CASE & POWER SUPPLY (192 WATT)
ASSEMBLED & TESTED 20MB ST225 \$325 30MB ST4038 \$625

ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA

IBM is a registered trademark of International Business Machines, CP/M, Concurrent DOS are registered trademarks of Digital Research Inc. MSDOS is a trademark of Microsoft. All above circuit board names are either registered trademarks or Trademarks of Viasys Corporation. CompuPro is a registered trademark of Viasyn Corporation. AT is a trademark of IBM.

NO EXTRA CHARGE FOR VISA OR MASTERCARD

COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 (313) 451-0665

**UPGRADE YOUR PC®, XT™, AT™,
OR BUILD A COMPATIBLE . . .**

FLOPPY DRIVES

Teac 55B 360K PC	\$106.95
Teac 55B 360K AT (Grey)	109.95
Teac 55GFV 1.2MB AT	139.95

TAPE BACKUP

20 MB Teac w/Contr & Soft	\$549
Streaming Tape Backup	

KEYBOARDS

At Style Keyboard	\$79.95
for XT or AT	

HARD DISK DRIVES

ST225 20 MB HD	\$328.95
ST4026 20 MB 40 MS	595
ST4038 30 MB 40 MS	625
ST4051 40 MB 40 MS	795
ST4096 80 MB 28 MS	1295
CDC 48/38 MB 28 MS	995
Toshiba 70 MB 23 MS	1495
CDC 86/72 MB 28 MS	1595
CDC 86/72 2 yr. Warranty	1795

CLONE CABINETS

5" Floppy or Hard Disk w/PS	\$99.95
AT Compatible Cabinet	99.95

POWER SUPPLIES

Skynet 135 for XT	\$89.95
Skynet 192 for AT	139.95

Most Reliable Power Supplies

We've Found!!!!!!

**CALL NOW
FOR LOW PRICES
ON
XT COMPATIBLE
&
AT COMPATIBLE
MOTHERBOARDS**

LASER PRINTERS

Canon 8 ppm 128K	\$1995
Canon 8 ppm 1.125	2995
Xerox 4045 (copies too)	4495

DOT MATRIX PRINTERS

NEC P6 PAR - 80 col	\$474.95
216/180/65 CPS	
NEC P7 PAR - 132 col	599.95
216/180/65 CPS	
NEC P5 XL Color PAR - 132 col	1195
290/240/100 CPS	

MONITORS

NEC Multi-syne for EGA Color	\$538.95
NEC Amber Monochrome TTL	124.95
Princeton HX12E EGA Color 12"	549.95
Princeton HX12 for Std Color	474.95

CIRCUIT BOARDS

**DFI — ONE OF THE WIDEST SOLD AND MOST RELIABLE NAMES IN CIRCUIT BOARDS
FOR PC, AT AND COMPATIBLES**

FD100	4 Floppy Controller	\$33.95	DIO100XT	Parallel, Serial, Clock	\$69.95
MIO100	Multi-I/O Floppy Card, par, ser, clock	88.95	DIO100AT	Parallel, Serial	73.95
MF100	Multi-Function Card, Parallel, Hercules™ Compatible	88.95	DIO200XT	Parallel, Serial, Clock Game	74.95
CG100	STD Color Card Fully Compatible with IBM	68.95	DIO200AT	Parallel, Serial, Game Port	78.95
CG200	STD Color Card with Parallel Port	79.95	MF3000AT	Parallel, Serial to 3072K w/OK	229.95
			MEMO2000XT	Includes Piggy Back Card Above Board Compatible EMS up to 2048K w/OK	189.95

Western Digital Floppy/Hard Disk Controller for AT	\$219.95
Everex AT Floppy/Hard Disk Controller	199.95
Vutek EGA 256K, CGA, EGA, MDA and Hercules™ Compatible	325
Mega 256K CGA, EGA, MDA Compatible	249.95

Vutek and Mega Enhanced Graphics Adapters Run with our 10MHZ at Compatibles

EFI Memo 2900XT above Board EMS Compatible with Drivers to 2048K with OK	\$189.95
--	----------

LOWEST PRICES ON ALL EVEREX EQUIPMENT

Internal Hayes Compatible 1200BPS Modem	\$139.95
---	----------

ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA — NO EXTRA CHARGE FOR VISA OR MASTER CHARGE
PC, XT & AT ARE EITHER TRADEMARKS OR REGISTERED TRADEMARKS OF INTERNATIONAL BUSINESS MACHINES.

COMPETITIVE EDGE

631 S. Main St. — Plymouth, MI 48170 — (313) 451-0665

INTRODUCING

VELOCITY 286-10™/12

A T™ COMPATIBLE



**10/12 MHz
OPERATION**

**1024K on MOTHERBOARD
16 BIT HARD DISK TRANSFERS
SERIAL, PARALLEL, CLOCK**

Enhanced Graphic Adapter Now Standard Runs with Color or Mono Monitor
1.2 MB Floppy, A T™ Compatible Keyboard, DOS 3.2

**30 MEGABYTE FAST HARD DISK
PRICES START AT \$2595**

DEALER & QUANTITY PRICING AVAILABLE

**BUDGET VELOCITY 20MB, DOS 3.2, 1024K, 1.2 FLOPPY
\$1995**

SHIPPING NOW!!!

— OPTIONS —

80287 Math CoProcessor

Enhanced Graphics Adapter

30 to 100 MB Hard Drives

20 to 100 MB Streaming Tape Back (2 minutes per 10 MB)

CPM®/Z80H Emulator Card Only **\$395.00**

8MHZ 8088 Slaves (up to 31) for Multi-User

3 MB Memory Expansion Only **\$795.00**

Monochrome & Color Monitors

Basic Xenix® 1024K, 8 Serial, 30MB System **\$4095**

NO EXTRA CHARGE FOR VISA OR MASTER CHARGE

CPM is a registered trademark of Digital Research Inc. Hercules is a trademark of Hercules. A T is a trademark of International Business Machines. Velocity 8 and are a trademark of Competitive Edge. Screen is PC Paint Brush copyright and product of 2Soft, Inc. Xenix is a trademark of Micro Soft.

Enhancing CP/M-80

by William F. Dudley

Add A Command History Function

Imagine that you are working at your CP/M system and you type the following command:

A>pip lst: + some.fil,another.fil, andmore.fil[t8p]

and instead of hearing the printer start, you get:

INVALID PIP FORMAT: + SOME.FIL,ANOTHER.FIL,ANDMORE.FIL[T8P]

A)

Wouldn't it be nice to get that last line back to correct the typo? It's so tedious retying the whole line again just for one small mistake.

Or imagine you are debugging a C program. Edit-Compile-Assemble-Link-Run-See errors. Edit-Compile-Assemble-Link-Run-See errors. Etc. etc. etc. ad nauseam.

It would be less tedious if you didn't have to retype the same five commands over and over again. (Even using a submit file, I still count three commands repeated endlessly.) Somebody *must* have thought of this before.

Well, campers, *they* did. Several big name operating systems have a command history feature:

a. Berkeley Unix. The cshell is several years old and it allows execution and editing of commands executed earlier. I find its syntax painful, however.

b. VMS 4.x. When Digital Equipment Corp. upgraded VMS, they added command history. They only allow access to the last twenty commands, which is sufficient for most needs. I think the editing characters are somewhat arbitrarily chosen, (unless you have a VT-100 terminal, where the arrow keys work).

c. AT&T UNIX V. A new shell for AT&T UNIX is called K-shell (Korn authored it). I find K-shell's command history very nice. It allows you to choose your command editor to look like either emacs or vi, both very popular editors on UNIX machines. It keeps the command history in a file, and stores many more than I need.

HOW DOES CP/M COMMAND HISTORY WORK?

To be useful for a CP/M environment, I worked up these requirements:

a. The command history is kept in (banked) ram to keep the speed to access commands high. I didn't want to slow down an already slow machine.

b. Sparse but useful feature set so that code size would fit within a standard CP/M system size. I still wanted to be able to use SYSGEN to make a bootable single-sided single-density 8" disk.

c. Use editor commands of an editor I was familiar with. I would have liked to use vi, but requirement (b) above dictated emacs, which was easier to implement.

d. Some trick features of the public domain CCP I was using had to be sacrificed to make room for history. These, plus other little used features and commands that history needed, are or could be supported by transient programs.

The resulting history feature works in the following manner. When faced with the CP/M prompt:

B)

you can type in a command just as before; however, the command line editing characters have changed and expanded:

[^]H destructive backspace (delete left)
DEL destructive backspace (delete left)
[^]U discard and erase entire line
[^]X discard and erase entire line
[^]B non-destructive move cursor left
[^]F non-destructive move cursor right
[^]A move cursor to beginning of line
[^]E move cursor to end of line
[^]D delete character under cursor (+ close up line)
[^]L toggle printer slaved to console
[^]C warm boot CP/M if typed as first char on line
[^]M (carriage return) execute line
[^]J (line feed) execute line
[^]P discard command buffer, fetch previous command
[^]N discard command buffer, fetch "next" command (not implemented)

Otherwise: insert the char into the command line. If not at the end of the command, do a real insert (i.e. move characters to the right over to make room.) Control characters not listed above are echoed just as CP/M used to (e.g. ctrl-Y is echoed [^]Y).

So now, fixing the command line offered at the beginning of the article would look like this:

A>pip lst: + some.fil,another.fil, andmore.fil[t8p]

INVALID PIP FORMAT: + SOME.FIL,ANOTHER.FIL,ANDMORE.FIL[T8P]

A)>P

A>pip lst: + some.fil,another.fil, andmore.fil[t8p]

Move the cursor to the "+" character. Use [^]A to go to the beginning of the line. Type eight (8) [^]F's to move right to the "+". Type [^]D to delete the "+". Your cursor is under the "s" in some, and the line now looks like this:

A>pip lst:some.fil,another.fil, andmore.fil[t8p]

Type an "=" to insert it in front of the "some". Type a <cr> (carriage return) to execute the line.

A>pip lst:=some.fil,another.fil, andmore.fil[t8p]

To make the edit-compile-run loop a little easier, you now need type the commands only once each:

A>edit foo.c
editing . . .
A>submit cc foo.c
compiler runs . . .
A>foo
foo runs . . .

Now, to re-edit, just type three (3) ^P's:

```
A)^P  
A)foo^P  
A)submit cc foo.c^P  
A)edit foo.c(cr)
```

From now on, you need type only three ^P's to recover the next command in the cycle.

Some Transient Programs: Three transient programs are used to support history. INITHIST.COM clears the history buffer to the empty state. This is unnecessary, but nice. If you ^P back to before you started the current session, the garbage in the history buffer can make a small mess on your CRT screen.

GETHIST.COM copies the history buffer to the CP/M transient area so it can be saved with the CP/M SAVE command. This command is good for debugging the history feature, but mainly used with:

PUTHIST.COM to allow keeping the history buffer on disk from one session to the next. This way, you need type a command only once, ever, ever, ever. (Well, almost).

SYSTEM REQUIREMENTS

The code presented here can be used with little change on CP/M 2.2 systems using a Z80 or 64180 CPU. I use a 64180, so the memory management instructions reflect that. However, these can be easily changed to accomodate a typical Z80 bank select system.

The second requirement is some free ram that CP/M doesn't use. This can be (as in my system) another 64K bank, or in the simplest instance, just a couple of pages of ram above the BIOS.

A third requirement is an assembler, and some experience SYSGEN'ing CP/M systems. I use SLR180 by SLR Systems in Pennsylvania. Microsoft's M80 and L80 can be used just as well.

WHAT PARTS OF THE O.S. ARE MODIFIED?

Hold onto your hats. This job is most cleanly done by modifying both the CCP and the BDOS. Yes, I said BDOS, but its not as bad as it sounds. Honestly.

In order to shoehorn all this code into a nearly stock CP/M system, I had to scrap the BDOS system call 0AH, Get Line from CON:. The history line editor replaces this code completely, and there was no sense in wasting the ram occupied by the line editor built into the BDOS. There are two approaches to the BDOS mod: the patch and the gung-ho version. More about this later.

The actual command history buffer management is handled by the modified CCP. I have been using an old public domain CCP replacement for years, (like ZCPR1 or something). I gutted out some functions in it that I never used to make room for the history buffer management code.

INSTALLING THE CODE

Modifying the BDOS: The BDOS change is the first to go in, and gives you a good command line editor right away, without fixing the CCP.

I disassembled my BDOS, and then made room by changing most of the absolute jumps to relative jumps. When I then excised the existing Get Line function code, I had just enough room for the new line editor (Listing 1). I also needed to add a second entry point for the getline function, to allow editing of a line already in the line buffer. I stole the unused CP/M function 26H. This function has never been used, in CP/M 2.2, CP/M 3.0 (Plus), or even MS-DOS (which copied CP/M's BDOS function calls.)

Public domain disassemblers are available to disassemble the BDOS, or you can buy the product advertised as a source code generator for CP/M and do it automatically.

If disassembling the BDOS seems like too much work, the other way is to patch the BDOS to call an external routine that you locate in high ram above the BIOS. This could be loaded at cold boot time or kept in a BIOS rom that is resident at all times.

To facilitate the patch technique, some useful locations in a standard CP/M 2.2 BDOS are shown in Listing 2.

Modifying the CCP: First, you need the source to your CCP, either the disassembled Digital Research CCP, or one of the public domain CCP replacements. I use an old public domain CCP. (It's so old, it doesn't have wheel bytes or named directories).

Next, you need to decide which features you will sacrifice to make room for the history code. I took out LIST (TYPE to printer) since I always use PIP for that, and JUMP (to start execution at an arbitrary memory location) since I use DDT if I need something like that.

Now the code in Listing 3 can be merged with your CCP source. It fits in just after the code to print the CP/M prompt ")", and replaces the code from there until the code to convert the command line to upper case.

As an aside, I also added code to toggle on/off the lower to upper case conversion of command lines. This allows you to use the PIP Start and Quit flags with lower case strings.

The only hardware dependent parts of the code are the pointers to where the history buffer is kept in memory. If you have a 64180 and more than 64K, then the code will run as supplied. I keep a copy of the CCP and BDOS from 10000H to 115FFH for warm boots from ram. The history buffer is kept from 11600H to 12000H. Subroutines BANK0 and BANK1 do what you'd expect, respectively disabling and enabling the memory from 10000H to 1CFFFFH. When enabled, that memory resides at logical addresses 0000H to 0CFFFFH.

If you have only 64K of ram, another possibility is to configure your system size to 60K, which will leave some ram above the BIOS that CP/M doesn't know about. The history buffer can be kept there just as well.

After you have made your decisions concerning history buffer location, modify your source code and get it to assemble without errors. Then make up a new system with DDT:

```
A)SYSGEN  
SYSGEN VER 2.0  
SOURCE DRIVE NAME (OR RETURN TO SKIP)  
SOURCE ON A, THEN TYPE RETURN  
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)(cr)  
  
A)SAVE 38 CPM.COM  
A)DDT CPM.COM  
.INEWCCP.HEX  
.r[offset]  
.INEWBOS.HEX  
.r[offset]  
.^C  
A)SAVE 38 NEWCPM.COM  
A)SYSGEN NEWCPM.COM  
SYSGEN VER 2.0  
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)B  
DESTINATION ON B, THEN TYPE RETURN  
DESTINATION DRIVE NAME (OR RETURN TO REBOOT)(cr)
```

Now the moment of truth. If you now reset and cold boot off your new system track (on the B diskette in this case) and get the A) prompt, congratulations. If "DIR" still works, a standing ovation. If ^P(cr) does a DIR again, then it's ticker tape parade time.

For the rest of you, I would suggest putting the BDOS mod in without changing the CCP right away. Everything should still work like a box-stock CP/M system, without history, but with a really fine command line editor facility.

Once the new BDOS (of BDOS patch) is in, the new CCP can go in. In order to help debugging the CCP mods for history, you may want to put in the transient programs GETHIST, PUTHIST, and INITHIST. (Listings 4, 5, and 6).

Another debugging trick for this is to temporarily put the history buffer into ordinary transient area ram, say at 8000H. This will allow you to examine the history buffer with DDT without having to worry about the complication of dealing with banked ram with DDT.

HOW IT WORKS

GETLINE function: The line editor in the BDOS follows closely the model of the original Digital Research code. Relative jumps are used to keep the code size down (they are slower than absolute jumps but speed is unimportant here). Some cleverness was required to make the routines for deleting and inserting characters work properly on any CRT with backspace the only cursor positioning sequence allowed. The temptation to use CRT escape sequences was immense, but the fact that I own two radically different CRT's forced me to do it right.

Backspaces are used to move the cursor left. To move the cursor right, the characters on the line are retyped. To erase, spaces are output, and then backspaces used to recover the cursor position.

The normal entry point, BDOS function 0AH, works as before, that is, the line buffer is cleared before the first character is gotten from the console in routine.

The second entry point, BDOS function 26H, (formerly unused) prints the current contents of the line buffer on the CRT, leaving the cursor at the end of the line, and then goes for an input character.

The CCP needs to know how the line was terminated, for if a carriage return or line feed was typed, the line is to be executed. If however, a ^P or ^N was typed, the line in the buffer is to be discarded and the previous or next line in the history buffer is to be fetched.

For this reason, BDOS function 0AH and the new function 26H now return the last character typed to the caller in the Accumulator (and the L register). CP/M applications that don't need this information are unaffected since the accumulator is expected to be clobbered by the BDOS call.

CCP History Function: When the CCP is first entered, it calls CP/M function 0AH to get a command line. If the call returns with a value of carriage return or linefeed, the command in the buffer is to be executed. When a command is executed by the CCP, it is first put into the history buffer. The history buffer is designed as a combination fifo (first in first out) buffer and a doubly linked list. The exact layout is as follows:

cbuff0:	dw	len?h	;pointer to previous command length byte
len0h:	db	len0	;length of first command
com0:	ds	len0	;first command
len0t:	db	len0	;length of first command
len1h:	db	len1	;length of second command
com1:	ds	len1	;first command
len1t:	db	len1	;length of second command

To copy a command into the history buffer, first the existing commands in the buffer are moved up in the buffer to make room at the input end. The new command is then copied into the buffer beginning, with a byte representing its length at each end of the command copy in the buffer. Finally, the previous command buffer pointer is reset to point to the most recent command in the buffer.

The copying of the latest command into the history buffer is done so that the history buffer never expands beyond its defined limits. The limits must be on page boundaries.

When the CCP gets a return value of ^P from the BDOS getline call, it goes to the fifo for the previous command. The command pointer is always pointing to the length byte for the previous command in the history buffers. That command is copied to the CCP command line buffer, and the previous command pointer is incremented by the length byte plus two so that it points to the next command back in the fifo (back in time).

The CCP now calls BDOS function 26H to edit the command which is in the command line buffer. Then, depending on the return value, the CCP either saves and executes the command, or fetches the previous command for editing again.

BUGS & NOTES

The only known bug is that command lines with embedded control characters are displayed incorrectly when editing. This means that when editing those lines, it is necessary to keep the real cursor position in your head. Another way is to always move using ^A and ^F's, so that the line characters will be echoed properly. I wanted to use the code, and not be debugging it for the rest of my life, so I decided to just use the code this way. If anyone fixes this bug, I'd like a copy of the fix.

The ^N command to get to the next command in the history buffer was not implemented due to lack of space in the CCP. If you overshoot the command you want with an extra ^P, the only way to recover is to send an empty line with a carriage return to the CCP, which will reset the command pointer to the beginning of the buffer.

VMS is a trademark of Digital Equipment Corp.
M80, L80, and MSDOS are trademarks of Microsoft.
SLR180 is a trademark of SLR Systems.



Bill Dudley is a Member of Technical Staff at AT&T Bell Laboratories in New Jersey, where he designs hardware for Cellular Mobile Telephone Service. He has a BSEE and M.Eng. from Cornell University. When not hacking his CP/M system or justifying why he hasn't got an MS-DOG system yet, he can be found riding his motorcycle.

Listing 1

```
; GETLIN.MAC
; bdos getlin replacement
; entry point to allow editing of line already
; in the buffer.
;
; William F. Dudley
; Dec. 22, 1985
;
wboot0 equ 0
lf equ 10
cr equ 13
bs equ 8 ;delete previous char
ctrlb equ 2 ;emacs back char <-
ctrlf equ 6 ;emacs fwd char ->
ctrln equ 0eh ;emacs next line
; (returns to caller if typed)
ctrla equ 1 ;emacs begin line <<-
ctrll equ 0ch ;echo to 1st: (replaces ^p)
ctrlp equ 10h ;emacs prev line
; (returns to caller if typed)
ctrle equ 5 ;emacs end of line -->
ctrlr equ 0bh ;emacs erase to eol
ctrld equ 4 ;emacs delete char
ctrlc equ 3
ctrlx equ 18h
ctrlu equ 15h
del equ 7fh
;
; <<< system call >>> buffer input from console
;
; buffer format: mx,n,c,c,c,c....
; mx= size of buffer
; n= # of characters entered
; c= characters
; special characters:
; rub/del = remove and echo last character
; ctrla = emacs move cursor to beginning of line
; ctrlb = emacs move cursor left one character
; ctrlc = reboots system if typed at start of input
; ctrld = emacs delete character under cursor
; ctrle = emacs move cursor to end of line
; ctrlf = emacs move cursor right one character
; ctrlh = remove and do crt rubout
; ctrlj = (line feed) end of input
; ctrlm = (return) end of input
; ctrlr = toggle list device slave
; ctrlu = scratch input buffer after new line
; ctrlx = scratch input buffer & back up to
; start of line
;
code0a: ld hl,(userde) ;hl= buffer pointer
        ld c,(hl) ;c= size of the buffer
        inc hl
        push hl ;save pointer to # of characters
        ld b,0 ;clear character counter
        ld a,(curcol) ;get current column position
        jr cament ;common entry point
```

```

code26:
ld    hl,(userde) ;hl= buffer pointer
ld    c,(hl)      ;c= size of the buffer
inc   hl
push  hl          ;save pointer to # of characters
ld    b,(hl)      ;b=no. of chars in buffer already
ld    a,(curcol)
push  af
push  bc
call  tline       ;print the line
pop   bc          ;restore buffer size, line length
pop   af          ;restore cursor position after prompt
coment:
ld    (colsav),a  ;save it
ld    d,b          ;cursor position == end of line
cod0aa: push  bc
push  hl          ;save the character count
cod0ab: call  qconch  ;get a character from console
and   7fh          ;strip parity
ld    e,a          ;current char to e
pop   hl          ;restore buffer pointer
pop   bc          ;restore character count
cp    cr
jr    z,aavec     ;if end of input
cp    ctrlp
jr    z,aavec     ;if end of input
cp    ctrln
jr    z,aavec     ;if end of input
cp    if
aqvec: jp   z,cod0ag  ;if end of input
cp    bs
jr    z,bckspc
cp    del
jr    nz,delchk  ;if not delete
;here for back space
;^a= character count
bckspc: ld   a,d
or   a
jr   z,aavec     ;if buffer empty, forget it
call  bspcc
push  hl
push  de
movlin: inc  d
inc  hl
push  bc
ld   a,b          ;no of chars to the right of cursor
sub  d
ld   b,a
push  bc
push  hl
; print line to end of line omitting current char
call  tline
ld   c,' '
call  pchar
call  pchar
pop  hl
pop  bc
push  bc
inc  b
inc  b
; backspace up to original position minus 1
bslp: call  bspc
djnz bslp
pop  bc
ld   a,b
or   a
jr   z,bckext
ld   c,b
ld   b,0
ld   d,h
ld   e,l
inc  hl
; move chars in buffer to close up gap
ldir
bckext: pop  bc
dec  b
pop  de
pop  hl
aavec: jp   cod0aa
delchk: cp   ctrld
jr   nz,fwd      ;^d ?
;cursor is after last char in buffer
push  hl
push  de
jr   movlin
jr   nz,eolchk  ;^f ?
;end of line check will occur there
eolchk: cp   ctrle
jr   nz,back
ld   a,b
sub  d
jr   z,aavec
ld   a,1
jr   forward     ;^e ?
;print (move) 'a' chars forward
forward: ld   e,a
add  d
ld   d,a
push bc
ld   b,e
call tline
pop  bc
jr   aavec
back: cp   ctrlb
jr   nz,beginl  ;^b ?
ld   e,i
call backup
beginl: cp   ctrla
jr   nz,eraeol  ;^a ?
ld   e,d
; backup e char's
backup: ld   a,d
or   a
jr   z,aavec     ;at beginning, forget it
push bc
ld   b,e
backlp: call bspcc
djnz backlp
pop  bc
jr   aavec
eraeol: cp   ctrlk
jr   nz,tglist  ;^k ?
ld   a,b
sub  d
; no of chars to erase
jr   z,aavec
ld   b,d
push bc
ld   b,a
ld   a,(curcol)
push af
push hl
; print chars to move cursor to end of line
; save so current hl is final hl
; now go rubout
call tline
pop  hl
ld   a,(curcol)
ld   (colsav),a
pop  af
ld   (curcol),a
; now go rubout
jr   back over the stuff to be erased
cod0al:
tglist:
cp   ctrll
jr   nz,cod0af  ;if not ^l
;here to toggle the list slave
push hl
ld   hl,listsw  ;save hl-reg
ld   a,1          ;point to list slave switch
sub  (hl)
ld   (hl),a      ;toggle it
pop  hl
jr   aavec
cod0af: cp   ctrlx
jr   z,clrlin  ;restore hl-reg
cp   ctrlu
jr   nz,cod0an  ;loop back
;yes, ^x
;no, not ^u
;here for crt line clear
;clean stack
cod0ag: ld   a,(colsav)
ld   hl,curcol  ;check print column position
cp   (hl)
jp   nc,code0a  ;against cursor position
dec  (hl)
call rubout    ;and:
;do the crt rubout
jr   cod0ag    ;if end of rubout, start over
cod0al: push hl
ld   a,(colsav)
or   a
jr   z,abvec
ld   hl,curcol
sub  (hl)
ld   (colsav),a
cod0am: call rubout
ld   hl,colsav
dec  (hl)
jr   nz,cod0am
abvec: jp   cod0ab
;echo character to console
cod0an: call putcon
ld   a,b
sub  d
jr   z,eolchr  ;last char in line, no move needed
push bc
push de
ld   b,a
ld   c,a
push bc
call tline
pop  bc
push hl
inbklp: call bspcc
;save buffer pointer to end
;back up from end of line to
;insert position

```

```

djnzb inbklp
pop hl
ld d,h ;hl points to end of buffer now
ld e,l
inc de
lddr ;move rest of line out one char
pop de
pop bc
; place character in buffer
eolchr: inc b
inc d
inc hl ;bump buffer pointer
ld (hl),e ;install character
ld a,(hl) ;get the previous character
cp ctrlc
ld a,b ;a= count
jr nz,cod0ap ;if not abort
cp 1 ;check for first character
jp z,wboot0 ;if first character
cod0ap: cp c
jp c,cod0aa ;if not at end of buffer
; here on end of input
cod0aq: pop hl ;restore pointer to count
ld (hl),b ;return the count in buffer
ld c,cr ;send a return and exit
call pchar
ld a,e
jp retaf
; bspcc-print a backspace, 2 if current char is control char
; bspc -print a backspace
bspcc: call bspc
dec d
ld a,' ' - 1
cp (hl)
dec hl ;doesn't affect flags
ret c ;not control char, only one
; screen position
bspc: ld e,bs
; print char in e register
putcon: push bc
push hl
ld c,e
call pchrcon
pop hl
pop bc
ret
; print chars from current cursor position to end of line
tline: ld a,b
or a
ret z
tlinel: inc hl
ld e,(hl)
call putcon
djnz tlinel
ret
;end needed if code is to be assembled as standalone patch
; end

```

Listing 2

```

; useful locations in the bdos
; referenced to the BDOS start
; which is found in locations 6 and 7
; on "normal" origin 0 CP/M systems
;
code0ap equ bdos + 05BH
code20p equ bdos + 093H
codetb equ bdos + 47H
colsav equ bdos + 030BH
colsvl equ bdos + 30AH
curcol equ bdos + 030Ch
gconch equ bdos + 0FBH
listsw equ bdos + 30DH
pchar equ bdos + 148H
pchrcon equ bdos + 17FH
retaf equ bdos + 301H
rubout equ 7FH
userde equ bdos + 343H
userhl equ bdos + 345H
wboot0 equ 0

```

Listing 3

```

title 'enhanced z80 ccp for 2.x 3/18/82'
; William F. Dudley
; Dec. 06, 1985 - modified for last command memory.
;
; equates
include listing.4a ;global equates for history

```

```

; TUNE THESE FOR YOUR SYSTEM SIZE
ccploc equ 0d400h ;start of ccp in memory
;I had to put CCP's stack above my BIOS to make room for code
histk equ 0F7FAH ;tune this as your system allows
; STOP TUNING HERE
;
ctrlp equ 10h ;previous command line
ctrln equ 0eh ;previous command line
supres equ true ;suppress user 0 prompt (true)
spmpmt equ '=' ;submit prompt character
cpmpmt equ '>' ;keyboard prompt character
defusr equ 0 ;default user # (com files)
maxusr equ 15 ;maximum user number
biosiz equ 380h ;bios size (get protection)
wboot equ 0 ;cp/m warm boot address
udflag equ 4 ;user number in high nibble,
;disk in low
tbuff equ 80h ;default disk i/o buffer
tfcb equ 5ch ;default fcb buffer
; macro to use 64180 output instruction
; replace out0 references with normal Z80 out
; instruction if using a Z80 cpu
out0 macro port
db 0edh,39h,port
endm
cbar equ 0bah ;64180 common bank register
bbr equ 0b9h ;64180 base bank register
; input command line from user console
tbl: subkil ;erase $$.sub if present
call setud ;set user and disk
ld a,cpmpmt ;print '>' prompt
call conout
ld c,0ah ;read (possibly old) command line
; from HERE on is neat new stuff
rcmd equ $ - 1 ;allows changing of getlin entry point
ld de,mbuff
call bdos
push af
call bankl
pop af
ctrlp
jr nz,newcmd
ld hl,(hdptr)
ld b,0
ld a,(hl)
cp buflen
jr c,rb2 ; command is legal buffer length
rb2: ld a,buflen ; command is legal buffer length
ld de,cbuff ; dest will be ccp buffer (mbuff + 1)
ld c,a
inc c ; add 1 for length byte
ldir hl ; hl now points to end length byte
inc a,h ; skip length byte
cp high hbuffe
jr nc,pchk1 ; hl points past hist buffer end
cp high cbuff0
jr nc,pchk2 ; hl doesn't point below hist
; buffer beginning
pchk1: ld hl,cbuff0 ;past end of hstbuf, wrap to
; beginning
pchk2: ld (hdptr),hl
call bank0
ld a,26h
ld (rcmd),a ;patch so cpm line editor invoked
vrstrt: jp restrt
newcmd: ld a,(cbuff)
ld c,a
ld b,0
inc c ; buffer length + length byte
push bc ; length of command
inc c ; add 1 for trailing cmdlen byte
ld hl,hbuffl
ld or a
sbc hl,bc
push hl ; # of chars in histbuf to move
ld hl,hbuffe
push hl ; destinatin
sbc hl,bc ; source
pop de ; dest --> de
pop bc ; # to move --> bc
ldir de,cbuff0 ; dest is start of hist buffer
ld hl,cbuff ; source is ccp command buffer
pop bc ; get command length
push de ; save start of hist buffer
push bc ; save command length
ldir ldir ; copy command to hist buffer
pop bc ; get command length
ex de,hl ; hl-->end of command in hist buffer
dec c
ld (hl),c ; put at end of buffer image in hbuff

```

```

pop    hl      ; get start of hist buffer
ld     (hdptr),hl; put in hist buffer pointer
call   bank0
ld     a,0ah
ld     (rcmd),a;unpatch so normal cpm getlin invoked
; capitalize string (ending in U) in cbuff
cnvbuf: ld     hl,cbuff      ;pt to user's command
ld     b,(hl)  ;char count in b
inc   b        ;+1 to cover 0 case
cbl:  inc   hl      ;next char
ld     a,(hl)  ;get character
ucasev equ  $ + 1 ;allows patching out ucase call
call   ucase ;force upper case
ld     (hl),a ;store character
djnz  cbl   ;do rest of command line
ld     (hl),b ;store ending <null> (b=0)
ld     hl,cbuff ;set cmd ptr to 1st char
ld     (cibptr),hl
ret
; This ^^^ ret needs to be ahead of ucase for CASE to work
; convert char in a to upper case
ucase: cp     'a'   ;less than lower case a?
ret   c      ;yes
cp     'z'+1 ;greater than lower-case z?
ret   nc   ;yes
res   5,a  ;no, clear lower case bit
ret
; form:
case: ex     (sp),ix ;put exit address on stack
casflg equ  $ + 1
ld     a,0      ;overwritten, initialized to ucase
; mode
xor   1
ld     (casflg),a
ld     a,low(ucase)
ld     b,'1'
jr    z,caspat
dec   a      ;points to the return before ucase
dec   b      ;is a '0' now
caspat: ld    (ucasev),a
ld     a,1f
call  conout
ld     a,b
jp    conout ;restart CCP without login
; make the second bank active
bank1: ld    a,10h ;second 64k bank
jr    obbr
; make the first bank active
bank0: xor   a
obbr: out0  bbr
ret

```

Listing 4

```

title  'get history buffer into save image'
; gethist.mac
; to use:
;   A>gethist
;   A>save 13 histname.ext
; for z80 with simple bank select: write to port 0 to
;   change bank
;   1 to port 0 -> normal ram from 0000 to 7ffffh
;   2 to port 0 -> second bank from 0000 to 7ffffh
;   include listing.4a ;global includes for history
;   aseg
;   org   tpa
; copy real program up to NONBNK above bank selected memory.
ld     hl,mstart
ld     de,nonbnk
ld     bc,pend-pstart
ldir
jp    pstart
mstart:
.phase nonbnk
; phase pseudo-op causes assembler to generate
; code for different run location than load address
pstart:
ld     a,bank2      ;switch in second bank
out  (bank),a
ld     hl,hbuff
ld     de,empty ;copy history buffer to unbanked
ld     bc,hbuffl ;ram above this code
ld     bc,hbuffl
ldir
ld     a,bank1      ;switch back to first bank
out  (bank),a
ld     hl,empty
ld     de,tpabuf ;copy buffer image down to low
ld     bc,hbuffl ;memory for CP/M SAVE command
ld     bc,hbuffl
ldir
ret
;return to cpm without warm boot

```

```

pend:
empty: ds    1
      .dephase
;
end

```

Listing 4a

```

; listing.4a -- global equates for history
; these should be tuned for your system hardware,
; particularly the history buffer location
;
hbuff  equ  1600h      ;pointer to current hbuff line
hptr   equ  hbuff
cbuff0 equ  hbuff + 2   ;contents is length of most
                        ; recent command
hbuffe equ  2000h      ;last location of history
                        ; buffer + 1
hbuffl equ  hbuffe - hbuff ;length of history buffer
bank   equ  0           ;bank register
bank2  equ  2
bank1  equ  1
nonbnk equ  8000h      ;start of non-banked memory
; STOP TUNING
tab   equ  09h      ;horizontal tab
lf    equ  0ah       ;line feed
ff    equ  0ch       ;form feed
cr    equ  0dh       ;carriage return
false equ  0
true  equ  not false
;
wboot equ  0          ;cp/m warm boot address
;
bdos   equ  5          ;bdos function entry point
tpa   equ  100h      ;base of tpa
tpabuf equ  tpa + 100h

```

Listing 5

```

title  'put history buffer from save image'
; puthist.mac
; to use:
;   A>get 100 histname.ext
;   A>puthist
; or if using CCP without GET command
;   A>dit histnam.com
;   ^C
;   A>puthist
; for z80 with simple bank select
;   include listing.4a ;global includes for history
;
aseg
org   tpa
;
move real program to high memory above bank selected ram
ld     hl,mstart
ld     de,nonbnk
ld     bc,pend-pstart
ldir
jp    pstart
mstart:
.phase nonbnk
pstart:
ld     de,empty
; move hist buffer image to high ram above bank selected ram
ld     hl,tpabuf
ld     bc,hbuffl
ldir
ld     a,bank2      ;select the second bank
out  (bank),a
; move hist buffer image to proper place in banked ram
ld     de,hbuff
ld     hl,empty
ld     bc,hbuffl
ldir
ld     a,bank1      ;re-select the first bank
out  (bank),a
ret
; return to CP/M CCP without re-booting
pend:
empty: ds    1
      .dephase
;
end

```

Listing 6

```

title  'clear history buffer'
; inithist.mac
; to use:
continued on page 85

```

A First Look At The 80386

By Michael Weinreich

Last October, Intel announced their 32-bit microprocessor, the 80386. This is the next generation in the evolving 8086 family of microprocessors which includes the 8088 used in the IBM PC and XT, and the 80286 used in the IBM PC AT.

A review of the specifications for the 80386 shows greatly increased power over what we have seen before from Intel, and power that (depending on your viewpoint and your biases) is either almost comparable to, or reasonably in excess of, that of competing 32-bit microprocessors. The 80386 will have a great impact not only because of this power, but because it is a convenient upgrade path from the present IBM PC family and clones. Although the benefits of the 386 are somewhat tempered compared with what Intel would have us believe, they are still very impressive.

WHAT THE 80386 IS SUPPOSED TO BE

Perhaps the best place to start is with the best news of all, that is, the claims that Intel makes for their chip:

- The highest performing commercial microprocessor ever designed.
- Clock speed of 16 MHz.
- Computational performance of 3-4 MIPS with a 32MB/sec bus bandwidth. With an 80387 numeric coprocessor, it can achieve 1.8M Whetstones/sec.
- Hardware support for multitasking, multiuser operating systems.
- Excellent compatibility with the 8086 family, including the 8088 used in the IBM PC.

This all sounds great, but then

IS THIS BELIEVABLE?

It may in fact be the highest performing commercial microprocessor ever designed, where commercial means that it is a single-chip CPU priced low enough to use in garden-variety commercial desk-top computers. But a chip that has been designed doesn't do us any good. Rumors of initial commercial availability vary from mid 1986 to early 1987, and there are many difficulties which may be encountered to delay whatever schedule is intended. The claim of the 16MHz clock rate is similarly impressive but sufficient yield for commercial availability at this speed is as yet unproven.

What Will The 80386 Do For Us?

The claim of 3-4 MIPS is impressive, and is believable given the specs for the chip. But it depends on the availability of the chip at the promised clock rate, and a high-performance memory system as well. The floating-point performance given by the 1.8M Whetstone/sec rating depends on the 80387 numeric coprocessor, for which we have only a vague promise of sampling (or is it just announcement) in late 1986.

The claim of multitasking and multiuser capabilities is true, in that the 80386 has an excellent set of instructions and features that make it relatively easy for writing of an appropriate multitasking operating system. But users are not going to write their own. We will have to wait for one to become commercially available.

The 386 is indeed more compatible with the 8088 than the 286 is, and this will be the biggest key to its success. This is also what makes it a kludge. A competent chip designer with a clean piece of silicon could easily come up with a much more consistent and organized design, and perhaps even with greater capability as well, if he did not have to work around the existing architecture of the 8088.

WHERE DOES THE TRUTH LIE?

Although not yet in mass production, the 80386 has been sampled and is running successfully in early prototypes at various facilities. It is not unreasonable to expect commercial availability soon (perhaps even by the time that this article is printed in a few months). Given the high-speed CMOS technology being utilized, clock rates can be expected to be substantially higher than were achieved with earlier processors in the family, even if the stated goals are not immediately reached.

For those who are looking for floating-point performance, the delay in availability of the 80387 is unfortunate. However, an adequate stopgap measure will be to run 80287 numeric coprocessors on small daughterboards in the 387

sockets. There are also other alternatives available (such as the Weitek chip set) for less patient power-hungry users.

Even with the strange instruction forms, and general disorganization that have resulted from the requirement that the 80386 be compatible with the 8088, the overall design of the 386 is very impressive. The designers have achieved a good balance between compatibility and inclusion of new capabilities, a balance that they failed to achieve with the 80286. They have managed to achieve compatibility without measurably impacting on performance, and they have allowed for simultaneous (multitasking) execution of old compatible programs along with new strictly 80386 programs. The new capabilities of the 80386, summarized in Figure 1, may appear at first to be unusual choices, but on careful examination they are seen as the important capabilities for future desktop PCs.

HOW HAS THE 80286 FARED?

Before considering how the 80386 will do, let us review how the 80286 (and the IBM PC AT which uses it) has fared in the marketplace. Most people use a PC-AT solely as a souped-up PC, because PC-DOS and application programs are not modified to do otherwise. The 286 does allow multitasking in a protected environment, and memory management with up to 16 MB of real memory and up to 1 GB of virtual memory. But it does not run 8086 programs in this environment, does not provide protected mode for running 8086 programs, and does not allow switching back and forth between 8086 and 80286 programs (although this can be accomplished with the assistance of external hardware, such as is provided in the IBM PC AT). So users of PC-DOS have not seen all the advantages of the 80286, and will not until the operating system is upgraded in spite of the difficulties in doing so. Only recently have we heard predictions that the necessary upgrade (DOS 5.0) will be available sometime in 1987.

The 80286 also offered instruction execution in fewer clock cycles, and this is where it has proven most valuable. The IBM AT, although running the same operating system and applications code run by an 8088 and, although run-

ning only at a clock speed increase of 26% (6 MHz vs 4.77MHz), executes that code about four times faster. This is because of the fewer clock cycles needed for each instruction, and also because the memory bandwidth is 16 bits rather than 8 bits. This makes the IBM PC AT a very valuable PC clone. Most ATs and clones are now sold at 8 MHz, and some are now available at up to 12 MHz, making them even more valuable PC clones. But they are still just PC clones.

THE 80386 AS YET ANOTHER CLONE

So will the 80386 be used in the same way, simply as a souped-up PC running the same old software? It could be, but it would be crippling the 386 just as the 286 has been crippled, and we would not see much of the real power of the 386.

The 80386 could have a faster clock speed, but we are now looking at only a 33% increase over the new 12 MHz AT clones. And for a 16 MHz 386, we will either need a reasonably expensive (for a desktop PC) memory system, or we will have to allow the memory system to slow down the CPU. The 386 can execute some instructions in fewer clock cycles (especially multiply and shifting instructions), but its improvement over the 80286 is much smaller than the 286's improvement over the 8088. The expansion of the memory bus to 32-bit doublewords is impressive, but no present programs can use this feature directly, since there are no 32-bit load or store instructions in existing IBM PC code.

We may conclude that if an IBM PC clone were to be built with a 386, and used to run the same software with the same PC-DOS operating system, the improvement of speed over a fast 286 machine would perhaps be 50% to 100%. But the price of the machine would be much higher, perhaps comparable to the increase in power. The performance improvement is impressive, but probably won't make waves in the present microcomputing world. Advances by factors of two are only impressive these days if they come bundled with a lower price as well! Including an 80387 provides greater benefits, but only for floating-point applications.

WHAT THE 80386 CAN BE

But do not despair! There is hope! Intel has provided in the 80386 a new mode referred to as "protected virtual 8086 mode", a feature which should have been provided in the 80286. With this mode, the 386 is able to run old PC-DOS software, and also run new software taking full advantage of the 386, under the control of a multitasking operating system.

The appropriate operating system for the 80386 may be a very extensive upgrade of PC-DOS, and if it isn't it will probably be made to look like it was. After all, there

Bus Specific Networks NEW SOLUTION

Complete network solutions for the price of a single card.

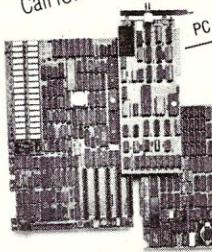
Entry level costs to networking are reduced by ICM's unique integration of the most common components of both fileservers and network workstations.

ICM's WS-286 contains all the common local area network components on one board—including ARCnet, SCSI, 1 MByte RAM, and 5 expansion slots. This greatly reduces system complexity, repair costs, and stock of multiple costly components.

The on-board SCSI controller provides easy interface to ICM's single-board disk and tape controller—so the WS-286 may be used as both a fileserv and inexpensive diskless workstation, supporting Novell and TurboDOS. It fits into standard IBM-XT enclosures without software or hardware modification. The 80286-based workstation operates at 10 MHz with no wait states when accessing on-board peripherals and RAM, while the expansion slots operate at 4.77 MHz to allow full XT expansion card compatibility.

Call ICM for more information and Reseller pricing.

PC, ARCNET, AND S-100 APPLICATIONS



ICM

TOMORROW'S BREAKTHROUGHS TODAY
InterContinental Microsystems

4015 Leaverton Ct., Anaheim, CA 92807
Phone: (714) 630-0964
Telex: 821375 SUPPORT UD
Easylink 62562040 BBS (714) 632-8750

are multitudes of users who have never used anything but PC-DOS (and would rather fight than switch). The multitasking capability will allow old PC-DOS software to run as tasks in the new virtual mode, and will also allow new 386 tasks to run with full utilization of the capabilities of the 386. This environment where old PC-DOS applications and new 386 applications can co-exist is the migration path to true 386 power. Users will at first use much of their PC-compatible software, while slowly introducing the new 386 software.

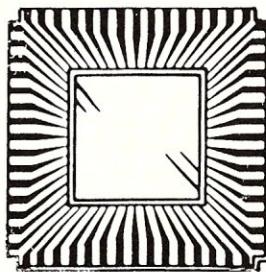
One of the most valuable advantages of the advanced OS will be the protection provided by the true multitasking capabilities of the 386. Tasks will be protected from each other, and the OS will be protected from the tasks. This may provide great safety, or it can allow riskier operation without sacrificing safety. For example, we can utilize a riskier disk caching algorithm, deferring storing

sectors from RAM onto the disk even until after an application crashes.

There is even compatibility for old PC-DOS applications that use the L/I/M Expanded Memory Specification (EMS). They will run without EMS memory, because the modified operating system can emulate the EMS standard using virtual memory paging.

Some old PC-DOS software may even be slightly modified to take advantage of some of the 386 capabilities, while still running in the virtual mode. For example, many programmers now purchase hardware debuggers, which assist them with aids such as trapping on access to data at a particular memory location, while the application otherwise runs normally. Some modification of software available with existing hardware debuggers will allow it to run using the 80386 internal hardware debugger, providing this feature at much lower cost than for existing systems.

While the user will still be able to run his old PC-DOS software, he will also be slowly converting over to new 386 software as it becomes available, and we may expect that the applications that can best take advantage of the power of the 386 will be the first to convert. The important features of the 80386 when running software written specifically for it are summarized in Figure 1. Many of them need no further explanation, but some deserve discussion:



8088 (IBM PC, XT)	80286 (IBM AT)	80386
8 and 16 bit registers	8 and 16 bit registers	8, 16, and 32 bit registers
8 bit bus	16 bit bus	32 bit bus
16 bit offset for addressing	16 bit offset for addressing	32 bit offset for addressing
Segment size 64 KB	Segment size to 64 KB	Segment size to 4 GB (gigabytes)
4 segment registers	4 segment registers Emulate 8088 only in real address mode, precluding use of many features of the 80286	4 segment registers Emulate 8088 in protected virtual 8086 mode, allowing many features of the 80386 to be simultaneously used
Memory 1 MB real	Memory 16 MB real 1 GB virtual	Memory 4 GB real 64 TB (terabytes) virtual
No memory management	Memory management including protection and virtual memory (with segments)	Memory management including protection and virtual memory (with segments and/or paging)
Single task	Task switches New instructions such as shift by immediate count, enter and leave procedure	Task switches More new instructions such as bit and bit string manipulation, jump on condition anywhere in segment, new addressing modes, more general use of registers
Clock speed 5 - 8 MHz	Faster instructions (fewer clock cycles)	Even faster instructions (fewer clock cycles)
Single step trap	Clock speed 8 - 12 MHz	Clock speed 12 - 16 MHz
8087	Single step trap	Hardware debugger, as well as single step trap
	80287 Task switching easier Emulate when missing	80387 Task switching easier Emulate when missing Enhanced trig function Faster (4x to 6x)

Figure 1. Summary of the features of the 8088, 80286, and 80386.

The 8086 family processors have partitioned memory into address spaces called *segments*. Segmentation is a valuable capability for memory management, but unfortunately it has gotten a bad reputation because of the limited size of the segments in the 8088 and 80286 (only 64 KB). Having increased the maximum segment size to 4 GB in the 386, this bad reputation should be forgotten. Meanwhile, the 8086 family has not utilized memory paging, a very valuable technique especially in a virtual memory system. Paging allows control of memory in small (in this case, 4 KB) *pages*. Both segmentation and paging are very valuable techniques, and neither is sufficient alone. Having both in the 386 is a great asset.

The 80386 includes, even more so than the 286 before it, a very valuable set of instructions and capabilities that make it very easy for a powerful true multitasking operating system to be written for it.

(These are hardware support for multitasking, and virtual memory paging). The 386 can therefore be expected to give superior multitasking performance, with less effort needed to write the multitasking operating system than would be required for other 32-bit microprocessors.

WHAT OS WILL WE USE?

We are not assured that a suitable OS will be available immediately, and at a reasonable price, to make available to us all of the benefits summarized above. We can of course continue to use our present version of PC-DOS, with the attendant limitations. We might also consider using DOS 5.0 if it is then available, but this is little better. It still limits us to running 286 code, barring us from the many added features of the 386, such as larger segments and 32-bit operations. Also, it keeps our old PC-DOS applications running in unprotected mode.

There have been many rumors, innuendos, and so forth telling us not to worry about running PC-DOS tasks on the 386. But every one of these that I have heard mentions UNIX (with PC-DOS subtasks) as the solution. I expect that this approach can be successful if it is available early, at a reasonable price, and does not require much re-education of the PC-DOS users. We have no assurances that it will have any of these characteristics.

Perhaps we can expect an intermediate sort of DOS. The modifications necessary to a PC-DOS operating system, to make it suitable for running on a 386, are about as difficult as what is now done by those companies offering multitasking under PC-DOS on present machines. Perhaps one of those companies will come forward with the solution. This would give us some benefits, such as multitasking with protection between tasks. This approach would provide greater benefits if applications developers could be convinced to provide upgrades of their products that could run as 386 tasks under such an OS.

We may then wonder what will happen to the IBM PC AT. The schedule for DOS 5.0 to unleash the power of the 286 is very close to the schedule for 386 hardware with its multitasking operating system. It will hardly seem worthwhile to upgrade the operating system, and get upgraded versions of applications, when an accelerator board with a 386 would allow much greater benefit with yet a different OS and different upgraded software. Perhaps DOS 5.0 will be a viable interim product, but it is likely to end up being an orphan product.

WHAT HARDWARE PRODUCTS WILL WE SEE?

One of the first products to appear will be 80386 accelerator boards for the IBM PC AT, and perhaps for the PC and XT as well. With a 386 alone, we will obtain some increase in power for old PC-DOS applications, and the advantages of the new operating system with its ability to run new 386 software with yet greater power. Provision of even an inexpensive cache memory on the accelerator board will provide substantially greater throughput. And when 80387's are available, very substantial improvements in floating point performance will be obtained.

There will also be desktop PCs provided with 80386's. Very few will probably attempt to harness the maximum power available from the 386, because of the costs involved. Considering that most of the advantages of the 386 are separate from the increased clock speed, I would not be surprised to see machines running as slow as 10 to 12 MHz. This will provide a majority of the maximum power at the lowest cost.

Another approach is to put additional expense into a cache memory that will allow the CPU to operate at near full speed, and then saving this expense by providing a very inexpensive memory system that would seriously degrade the 386 performance if the cache were not present. For example, the current IBM PC AT expansion cards could be used, providing 16-bit memory (half the 386 bus size) at 8 MHz (half the 386 clock speed) with a wait state. This not only provides very good average performance; it provides the best possible performance for benchmarks that can be used to advertise the machine.

LOOKING BEYOND THE 80386

Intel has already indicated that they are planning for the next-generation microprocessor (at least in between their hints that the 80386 is more than we could ever want for years to come). They tell us it will be called the 80486, but they provide very little in the way of details. But we may make some educated guesses (and in some cases, better than guesses, given the scraps of evidence available):

On-chip access time is always fastest, so on-chip caches can be very effective even when they are small. Therefore, there may be a code cache on the chip, but more likely there will be an enhanced instruction prefetcher which prefetches code along both paths following each conditional branch. There will likely be a data cache on the chip (perhaps 8 or 16 32-bit words, fully associative), and perhaps a stack cache as well. The stack caching could of course be handled within a data cache, but a stack cache is so much easier to implement that it may be worthwhile to keep it separate.

In order for Intel to get some more real-time controller and data acquisition business, they will have to provide faster interrupt response. This will, however, likely be a special form of interrupt, not applicable to speeding up most interrupts as we encounter their use in regular processing.

There will of course be even more parallel operation of functional units on the CPU chip, new instructions, faster instructions (in clocks), and even faster clock speeds (although clock speed will not be the most important way in which throughput will be enhanced).

There is some evidence from Intel that there will be 32-bit segment selectors, which might indicate an increase in the virtual memory size (is 64 TB not enough?). More likely this indicates that the OS will be modified to handle a greater multitude of tasks, which might occur in a multiprocessor configuration supporting a great many users. We might even go so far as to speculate that Intel may be planning to enter the mainframe market with such a multiprocessor configuration.

Bus Specific Networks

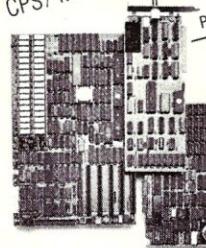
NEW FUTURE

Finally PC-DOS capabilities on S-100 systems—a major breakthrough the noncompatibility barrier.

For the first time, you can run monochrome PC-DOS and MS-DOS—including Lotus 123, Wordstar, and Sidekick—on your existing S-100 system. All with the brand new CPS-16F, a powerful single board network processor, only from ICM. It fits existing installations without a LAN hookup and has low entry requirements for TurboDOS Systems. The CPS/MS-1000 package (with 1000 opportunities) consists of the CPS-16F processor, M/STER—ICM's monochrome graphics PC-emulating terminal; PC-DOS; TurboDOS/PC; and ICM's exclusive MS-1000 software.

ICM brings PC-DOS to the S-100 industry. Call ICM for Reseller pricing on the CPS/MS-1000 package and info about our upcoming products.

PC, ARCNET, AND S-100 APPLICATIONS.



ICM

TOMORROW'S BREAKTHROUGHS TODAY
InterContinental Microsystems

4015 Leaverton Ct., Anaheim, CA 92807
Phone: (714) 630-0964
Telex: 821375 SUPPORT UD
Easylink 62562040 BBS (714) 632-8750

AND FURTHER INTO THE FUTURE

We will be ready for the next generation of hardware after the 80486 in about 5 years. At that time, our stock of 8086 family software will fall into three categories:

First, there will be old PC-DOS programs which were never converted to 286 or 386 code. This implies that performance considerations are not important. Such applications can be run under an emulator; i.e., a supervisor program that can inspect and effectively execute object code for a foreign machine. This is inefficient, but sufficient.

Second, there are the applications now available, and yet to be written, which are developed in high-level languages. These can easily be re-compiled for any new CPU architecture.

Third, we have the programs that have been converted to 386 code (or at least to 286 code), and which require too much computational power to run effectively under emulation, but which cannot simply be re-compiled. Luckily, these programs are generally well-behaved. The 286 and 386 environments frown upon techniques such as self-modifying code, and prohibit manipulation of segment register addresses without the cooperation of the OS. This promotes clean object code, compared with the dirty object code common in old PC-DOS applications. While

this clean code is being developed over the next few years, we may expect that other chip manufacturers will be copying many of the important features of the 8086 family, such as segmentation of memory with a few segment base registers as part of the CPU. This will leave us with an environment in which conversion of object code to run on a foreign machine will be feasible, especially given the incentive for competing manufacturers to develop better conversion aids in order to make 8086 family software available on their CPU architectures.

Compatibility with the 8086 family on a hardware level will therefore become relatively unimportant. Unnecessary compatibility will not be allowed to interfere with progress. We may expect that even Intel will not be overly concerned with 8086 family compatibility after the 80486. □

Michael Weinreich is an independent consultant in Arlington MA, who has extensive experience with machines from PCs to Crays in areas such as machine architecture, operating systems, language translators and numerical methods. He holds SB, SM, EAA, and ScD degrees from MIT. The first public presentation of this information was at the PC Technical Special Interest Group of the IBM PC Group of the Boston Computer Society earlier this year.

PC-PRO

by Alex K.H. Soya

Many S-100 computer owners feel left out these days because most new software is aimed at PC-DOS/MS-DOS systems. Today's computers must be able to provide some degree of PC compatibility to take advantage of software written for PC-DOS. Fortunately for owners of CompuPro S-100 systems, PC-DOS is no longer a foreign operating system.

PC-PRO, a Computer House product, is an implementation of IBM PC-DOS versions 2.10 and 3.10 designed to run on CompuPro S-100 hardware. PC-PRO is not based on a Microsoft generic version of PC-DOS (MS-DOS), but on the IBM specific version, as supplied by IBM, for the IBM-PC computer. The evaluation copy used for this review of the CompuPro implementation is based on the PC-PRO BIOS version 2.42 and PC-DOS 3.1.

INSTALLATION

PC-PRO is shipped with the original IBM DOS 3.1 manuals and diskettes as supplied by IBM. Also included are an 8" disk containing the operating system as implemented on the CompuPro, an EPROM (Erasable Programmable Read Only Memory) for the floppy-disk controller board, and the *PC-PRO User Guide*, a manual written by Computer House.

The user guide is well written and explains in detail how to use the utilities provided with PC-PRO to configure the operating system for a particular hardware environment. Hardware switch settings for some of the I/O boards are also briefly discussed. The manual gives several explanations as to why certain programs will or will not run with PC-PRO. Though earlier versions of PC-PRO included much of the source code for the BIOS, version 2.42 includes only some minimal parts of the source code.

Before the operating system can be booted, the supplied new EPROM, containing a special boot loader, must be installed on the DISK1 or DISK1A floppy-disk controller card. Some switch settings must also be changed on the DISK1A controller as the supplied EPROM is a 24-pin device rather than a 28-pin EPROM as supplied by CompuPro originally. The manual states that the EPROM must be installed in the lower 24 pins of the socket, however a diagram may make this point more obvious and

Running PC-DOS 3.1 On A CompuPro S-100 System

easier for those not familiar with handling chips on a daily basis.

Unfortunately the routines in the EPROM no longer allow the system to boot from the DISK3 hard-disk controller. I usually run the Concurrent DOS 4.1 operating system and boot directly from the hard disk. With the replacement EPROM, I had to boot from a floppy disk. Concurrent DOS 4.1, CPM-80, and CPM-86 will boot normally from the floppy.

To boot the out-of-the-box PC-PRO configuration, it is necessary to have a console connected to the serial port of the System Support Board. The system booted at the first attempt, which delighted me as I usually have to experiment with different switch configurations when installing new operating systems.

The first thing to do, after booting PC-PRO, is to make a backup of the master disk by executing a provided batch file. The batch file will format a new disk and copy the entire master disk, including a bootable system. This operation is straight forward and almost automatic.

CONFIGURATION

The next step in bringing up PC-PRO is to configure the operating system for the machine's particular hardware environment. A utility program called CONFIGIO.EXE is supplied for this purpose. CONFIGIO is an interactive program which allows the user to configure character devices such as terminals, printers, communications ports, floppy-disk drives and controllers, clock support and timer, interrupt controllers, IBM-PC ROM BIOS emulation, and some other miscellaneous options.

Character devices can be configured according to baud rate, stop bits, parity, and handshaking parameters. In addition, port addresses, bit-masks, initialization bytes, device redirection, and interrupt specifications can be selected. Six different I/O boards are currently supported: CompuPro System Support 1, CompuPro Interfacer 3 or 4, CompuPro Interfacer 2, Lomas Hazitall, Macrotech ADIT I/O, and Fulcrum VIOX-2. Other boards could be installed by selecting an *unspecified* type and defining the various port addresses, masks, and initialization bytes.

Floppy-disk drives and controllers can also be selected with a wide range of parameters for various types of drives. I used this option to install a 5 1/4", 48-TPI drive on the DISK1A floppy-disk controller. Both 48 and 96-TPI, 5 1/4" and 8" single or double-sided drives are supported. Also the new IBM-AT high capacity 5 1/4" drives rotating at 360 RPM, rather than 300 RPM, can be used with PC-PRO. For each of the drives, it is possible to select the various drive parameters such as seek step rate, head unload time, and head load time. Two floppy-disk controllers are supported by the standard BIOS: CompuPro DISK1A and Lomas LPD72. Other options allow the user to enter the port addresses to locate the controller at various I/O ports.

Clock support and timers can be configured to use either the internal tick count or the System Support clock to keep the time and date. Selecting the System Support clock enables PC-PRO to boot with the correct time and date, whereas using the tick count option forces the user to enter the current time and date when booting the system. Again, port addresses and other hardware dependent values can be selected.

Interrupt controllers can be installed and their parameters modified to suit particular hardware configurations. As with the other devices, port locations and initialization bytes may be user defined.

The IBM-PC ROM BIOS emulation is performed in the PC-PRO BIOS and can be configured to handle various serial terminals. The BIOS emulation allows programs, which control the various screen attributes through software interrupt calls, to handle various serial terminals. Computer House points out that the emulation

From the Editor:

Owners of CompuPro systems who are running MS-DOS or Concurrent-DOS, or are planning to run these operating systems, will find it advantageous to join the C-PRO National User Group, Box 2146, Woodbridge VA 22193. They publish a quarterly journal containing a great deal of information for CompuPro system owners, many of whom are now running MS-DOS and Concurrent. The most recent copy of their journal carried an excellent article on modifying the CompuPro PC video card to get Hercules high resolution and IBM color graphics compatibility.

is by no means perfect and thus makes no assurance that it will function correctly. This did not bother me, as I was using the CompuPro PC-VIDEO board rather than a serial terminal.

CONFIGIO is one of the most versatile hardware configuration utilities I have ever encountered on a microcomputer system. With a bit of thought and planning a wide variety of configurations could be supported. Though some of the options are rather technical and require knowledge of the hardware, the technical nature of these options is what makes the CONFIGIO utility so useful.

HARD DISK DRIVES

PC-PRO supports hard disks running with the CompuPro DISK3 and DISK2 hard-disk controller cards. The drivers for the DISK3 are built into the BIOS and are thus always present. The DISK2 drivers must be read in at boot time as specified in the CONFIG.SYS file. As I do not have a DISK2 controller, the following pertains only to the DISK3 card.

A utility called HDMAINT is provided to partition the hard disk into DOS and CP/M media. DOS-media formatted partitions are available to PC-PRO. CP/M partitions are available to either CP/M-86 or Concurrent DOS-86. HDMAINT allows the user to specify which tracks on the hard disk are to be partitioned as DOS media. For those who intend to use only PC-PRO as their operating system, the entire hard disk may be used as DOS media. However, if the hard-disk drive is to accommodate two operating systems, as in my case, only part of the drive may be used by PC-PRO.

HDMAINT also allows the user to specify the structure of the directories used on the DOS partition. As I already had DOS partitions on my hard disk from Concurrent DOS 4.1, I hoped to be able to match these partitions and thus share the DOS media drives between the two operat-

Bus Specific Networks

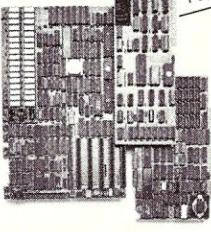
NEW SIZE

4-in-1 processor board.

More compact, our new CPS-Q6A opens a myriad of new opportunities. Only from ICM. Combining four 8-bit network processors on a single board shrinks the bus space, uses 35% less power and costs less per processor. Designed to operate at 10 MHz, it offers S-100 users the highest degree of performance and system integration available in the market today!

ICM's CPS-216 is also available—with two 16-bit network processors operating at 8 MHz with 256K RAM each, all on a single board. ICM provides the system integrator with the best price performance products possible. Call ICM for Reseller pricing on the CPS-Q6A and CPS-216 package and info about our upcoming products.

PC, ARCNET, AND S-100 APPLICATIONS.



ICM
TOMORROW'S BREAKTHROUGHS TODAY
InterContinental Microsystems
4015 Leaverton Ct., Anaheim, CA 92807
Phone: (714) 630-0964
Telex: 821375 SUPPORT UD
Easylinx 62562040 BBS (714) 632-8750

ing systems. With the aid of the DOS-media DPBS from Concurrent DOS 4.1, I attempted to duplicate my drive configuration to allow PC-PRO to access the existing DOS partition. Unfortunately, the attempt destroyed the DOS partition instead. PC-PRO does not seem to know about Concurrent DOS 4.1 DOS-media support and assumed the drive to be CP/M media. After installing the DOS partition using HDMAINT under PC-PRO, I was able to modify the DOS-media DPBS of Concurrent DOS such that I could create a DOS partition which could be read by both operating systems.

HDMAINT is a well thought out program and is of great value when installing DOS partitions. It is essential, however, to backup any previous files on the hard disk as they could easily be destroyed if a mistake is made while assigning partitions.

The DISK3 hard disk performs very well with PC-PRO. The access speed is very fast and smooth. Commands seem to load and execute almost immediately. The performance improvement compared to an IBM-PC is very noticeable. Though the manual does mention support for booting from the hard disk, a special ROM is required which was not available at the time of this review.

OTHER UTILITIES

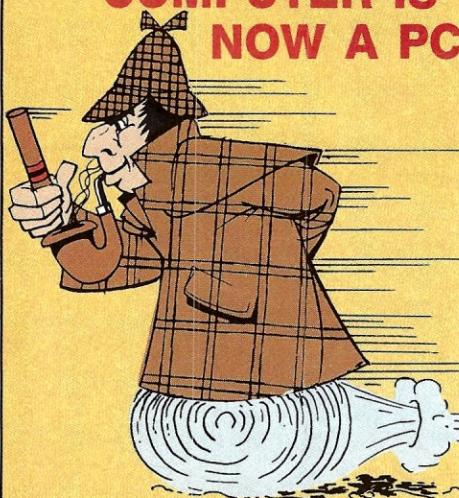
Computer House provides some other utilities with PC-PRO. DISKBPB is used to display the Bios Parameter Block (BPB) of a specified floppy or hard-disk partition. REFORMAT allows the user to reformat the hard disk without destroying any of the existing data. The major reason for using REFORMAT is to apply a new skew on the hard disk to take advantage of the maximum possible speed of the DISK3 controller. The new format is transparent to all existing software.

HDVERIFY is used to verify the integrity of the data on the hard disk controlled by the DISK3 card. This utility reads every track on the hard disk and reports any errors. The verification is noticeably fast so that the entire hard disk can be verified in a few minutes.

With FIXBOOT, the system can be booted or rebooted from the hard disk. This utility requires a boot ROM which is capable of booting from the hard disk.

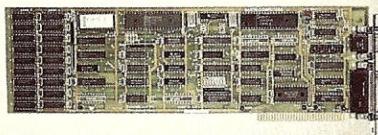
Instead of the IBM-PC format utility, Computer House supplies its own FORMAT program. This utility supports single and double-sided 8" and 5 1/4" disks. For the 5 1/4" disks, both 48 and 96-TPI drives are supported as well as the IBM AT 360 RPM high-density (1.2M) drive.

THE WORLD'S FASTEST Z-80 COMPUTER IS NOW A PC



EARTH COMPUTER's TURBOSLAVE-PC™ is the world's fastest Z-80 Coprocessor. Running at 8MHz, it was designed to permit operation of thousands of CP/M application programs on your IBM-PC, XT, AT™, or compatible computer system.

The TURBOSLAVE-PC supports the TurboDOS™ multi-user operating system which allows up to 16 users on your PC. It is the only IBM-PC/Z-80 system that is MP/M™ compatible and allows TRUE multi-user, multi-process operations, including full record locking and security.



Discover a whole new world of high-speed (8MHz) single and multi-user applications for your personal computer. Discover the TURBOSLAVE-PC... the world's fastest Z-80 Coprocessor, with such outstanding features as:

- 128K RAM with parity
- 2 Serial ports
- On-board Counter Timer
- S.L.R. Z-80 assembler included

To order your TURBOSLAVE-PC, call or write to:



EARTH COMPUTERS

P.O. Box 8067, Fountain Valley, CA 92728
TELEX: 910 997 6120 EARTH FV

(714) 964-5784

Ask about EARTH COMPUTERS' other fine PC and S-100 compatible products.

IBM-PC, XT, AT are trademarks of International Business Machines, Inc.; CP/M and MP/M are trademarks of Digital Research; TurboDOS is a trademark of Software 2000; TURBOSLAVE-PC is a trademark of Earth Computers

A PC-PRO version of the MODE command allows dynamic configuration of various serial devices. VOLUME can be used to change the volume label on a diskette. A very useful utility for those who are changing from CP/M to PC-DOS is INTERDOS. With this utility, users can transfer files from CP/M-media to DOS-media floppy disks. Supported CP/M media disks include 8", 128-byte sector and 1024-byte sector disks with CompuPro format, as well as 5 1/4" disks as per IBM-PC Concurrent CP/M specifications. All of the above utilities work well and are easy to use.

PC-VIDEO SUPPORT

To install routines which support the CompuPro PC-VIDEO board it is necessary to edit the CONFIG.SYS file with the specifications of the device driver containing the PC-VIDEO routines. The device driver, VIDEO.SYS, is read into memory and executed when the system boots and provides support for standard video I/O and various ROM BIOS interrupt calls.

The PC-VIDEO support is limited by hardware compatibility with the PC-VIDEO card. Computer House states that programs which access either IBM-PC timers or sound generation support may not function correctly. There also seems to be a compatibility problem with the keyboard port location which could be overcome by a hardware change. Since many public domain programs access IBM-PC hardware directly, they may not run correctly or might hang up the system. Pressing CTRL-ALT-DEL causes a special interrupt routine to be executed which prompts the user to either resume running or to abort the program and return to the system prompt. This feature is very useful in recovering from hung programs without having to reboot the entire system.

The PC-PRO User Guide lists the following programs as known to work correctly: Lotus 1-2-3, SuperCalc 2 and 3, dBase-II, File Master (Schuler and Associates), Mainstreet Filer (MainStreet Software), Wordstar 3.2, 3.3 and 2000, PC-WRITE (Quicksoft), Manx Software's C Compiler and the Z editor, List (public domain, by Vernon D. Buerg), and ANSI.SYS from the IBM master diskette.

I tried many graphics-oriented, public domain programs with the PC-Video card under PC-PRO. While some ran without any problems, others did not run correctly when either sound was generated or access to particular hardware ports was attempted. I did notice that the screen image on a color monitor was shifted to the left when in high-resolution graphics or 40-character alphanumeric mode. This is not the case if the same programs are run under Concurrent DOS 4.1, suggesting there

PC-PRO PRODUCT INFORMATION

Manufacturer

Computer House, Inc.
P.O. Box 709
Woodacre, CA 94973
(415) 453-0865

Price

\$395

may be a problem in the PC-VIDEO driver routines.

I also noticed that screen scrolling is accompanied by an annoying flash on the upper quarter of the image. I have observed this flash on the IBM-PC; however, it does not occur when running Concurrent DOS 4.1.

SYSTEM BEHAVIOR

In general, the system behaves very well. Floppy disk accesses are noticeably fast, compared to CP/M-86 or Concurrent DOS 4.1. Most of the programs I ran worked fine. Care must be taken to avoid compiled programs which try to access the BASIC ROMS on an IBM-PC or, as already mentioned, try to manipulate hardware directly.

I was disappointed to find that TURBO PASCAL Version 2.0 by BORLAND, as implemented for the IBM-PC, would not run the provided sample program, ART.PAS. The program crashes with the message *Unexpected interrupt 01h trapped* followed by a register dump. This program works fine under Concurrent DOS 4.1. Again, this indicates a problem with the Video drivers rather than the hardware.

SUMMARY

PC-PRO by Computer House is an excellent implementation of PC-DOS on the CompuPro S-100 system. The configuration utility, CONFIGIO, allows the system to be tuned over a wide range of hardware selections with relative ease. Support is provided for various types of floppy disk, hard disks, and the CompuPro PC-VIDEO board. All of the supported devices worked flawlessly.

If an application can run on an MS-DOS generic system, it should perform well with the PC-PRO system. Programs written specifically for the IBM-PC should be tested carefully before purchasing to ensure that they will function correctly. Copy protected programs require a 5 1/4" floppy disk to load and even then may not load as they use other hardware specific items, such as timers, in their copy protection procedures.

Considering that the architecture of the CompuPro system is completely different than the IBM-PC's, Computer House has performed an excellent job in porting

PC-DOS. In future releases I would like to see a new version of the EPROM so that my usual operating system can boot directly from the hard disk. I would also like to see some improvements in the PC-VIDEO drivers to solve the non-hardware related problems.

As We Go To Press

After this review of PC-PRO was completed, and set in type, Computer House sent me a new boot ROM (Version V4-1A) for the DISK1A controller and a updated BIOS version 2.50.

The new ROM allows booting of 8" and 5½" floppy disks, as well as the DISK3-controlled hard disk drive. The new ROM is similar in operation to the original ROM supplied by CompuPro. However, I did experience some difficulty in booting the system with a Macrotech MI-286 CPU board. The system would ignore the floppy disk controller and always try to boot from the hard disk. This problem did not appear with the older CompuPro 85/88 CPU card. The MI-286 would boot fine with the original CompuPro ROM. The difference between the ROMs, according to Computer House, is that PC-PRO boots from a double density 8" disk, while CompuPro boots Concurrent CP/M from a single density track.

It seems that the higher density selection during the boot cycle is more sensitive to noise on the bus with faster CPU speeds. In any case, the new ROM is invaluable if a hard disk is attached to the system.

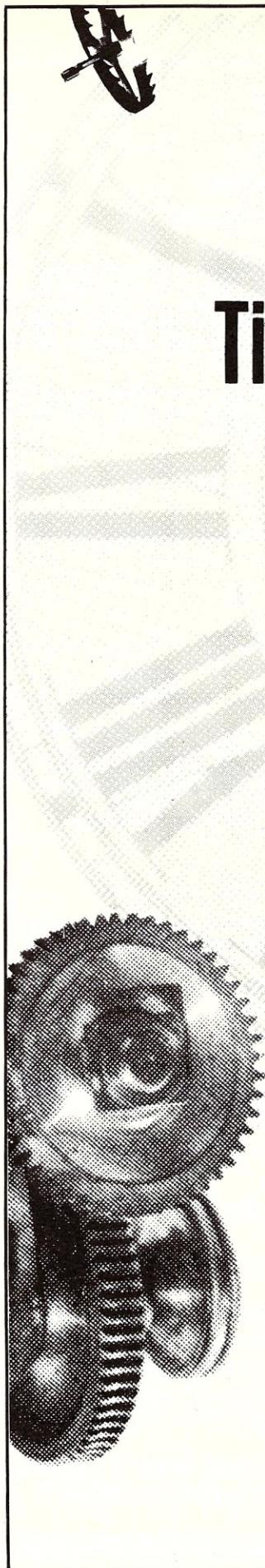
The new BIOS also fixes several bugs and introduces some new features and enhancements. The serial drivers have been enhanced to allow more flexible use of dumb terminals. It is now possible to map function and control keys to IBM-PC keyboard codes. This feature allows software, which expects IBM-PC keyboard codes, to run with ordinary serial terminals.

A "Visual Bell" has been added to the PCVIDEO drivers to make up for the absence of any bell circuitry on the PCVIDEO card. When a bell character is sent, the screen flashes brightly to alert the operator.

Other enhancements and bug fixes have been made to the video drivers, the MODE command, HDVERIFY and M-Drive/H drivers.

The Turbo Pascal graphics demo still crashes with the message "unexpected interrupt 06h" followed by a register dump.

Alex Soya has been involved with CP/M for over 9 years. His interests include operating systems, data communications, system utilities, real time and scientific programming. He is currently a senior in computer science at Florida Institute of Technology. [square]



BD Software, Inc., maker of the original
CP/M-80 C Language Development
System, knows

Time is precious

So the compilation, linkage and execution speeds of BDS C are the fastest available, even (especially!) on floppy-based systems. Just ask any user! With 15,000+ packages sold since 1979, there are lots of users . . .

New! Ed Ream's RED text editor has been integrated into the package, making BDS C a truly complete, self-contained C development system.

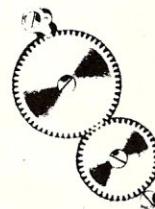
Powerful original features: CDB symbolic source-level debugger, fully customizable library and run-time package (for convenient ROM-ing of code), XMODEM-compatible telecommunications package, and other sample applications.

National C User's Group provides direct access to the wealth of public-domain software written in BDS C, including text editors and formatters, BBS's, assemblers, C compilers, games and much more.

Complete package price: \$150.
All soft-sectored disk formats, plus Apple
CP/M, available off-the-shelf. Shipping: free, by
UPS, within USA for *prepaid* orders. Canada: \$5.
Other: \$25. VISA, MC, COD, rush orders accepted.

BDS Software, Inc.

BD Software, Inc.
P O Box 2368
Cambridge MA 02238
617 • 576 • 3828



BSW-Make

A practical and efficient
software configuration manager
for MS-DOS, VAX/VMS, and VM/CMS

At The Boston Software Works, we routinely work with a number of different operating systems and development environments. One tool we have found to be indispensable is BSW-Make. BSW-Make is a complete implementation of the UNIX *make* utility. It automates the tedious task of rebuilding your software product after an editing session; BSW-Make does only the minimum work required to update your product after a change, saving time and preventing missed compiles.

We carefully constructed BSW-Make to be portable, and have used it successfully under MS-DOS, PC-DOS, VAX/VMS, and VM/CMS. We wouldn't want to start a major software project without it, and we think you won't either, once you've tried it.

Highlights of BSW-Make:

- Works with any compiler, assembler, linker, or text processor
- Not copy protected
- Indirect command file generation facility overcomes operating system command length limitations
- Macro facility for parameterized builds
- Syntax compatible with UNIX *make*
- 30-day unconditional money-back guarantee

MS-DOS
\$89.95

VAX/VMS
\$495.00

VM/CMS
\$550.00/yr

BSW-Make for MS-DOS runs on any MS-DOS machine. It requires MS-DOS or PC-DOS version 2.00 or later, and is shipped on IBM PC 5 1/4 inch diskettes or IBM PC Convertible 3 1/2 inch diskettes.

(Available September, 1986)
BSW-Make for VAX/VMS runs on any VAX or MicroVAX running VMS version 4.0 or later. It is shipped on 9-track magtape or RX50 diskette.

BSW-Make for VM/CMS runs on any IBM 370-series, 43xx, 308x, or 309x system running VM/CMS. It is shipped on 9-track magtape.

All prices include shipping within the United States. Foreign orders add \$10.00 shipping and handling. We accept checks, MasterCard or VISA, or company purchase order.

The Boston Software Works, Inc.
120 Fulton Street, Boston, MA 02109
(617) 367-6846

PC-PRO is PC-DOS

for
CompuPro/Viasyn

- PC DOS Ver 3.1
- PC Graphics Support

for information write

Computer House, Inc.

P.O. Box 709 Woodacre, CA 94973
(415) 453-0865
Trademarks: Viasyn Corp.,
PC-PRO - Computer House.

S-100 BARE BOARDS

8086/8087 CPU - plus
2764 or 27128, 8253, 8259

8088 Auxiliary Processor
I/O mapped, 4K EPROM,
4K RAM, prototype area

**\$45.00
Each**

Call or write for brochure.

Terms: Check or money order only. CA residents add sales tax. Prices include UPS shipping.

Applied Innovations
3000 Scott Blvd. Suite 106
Santa Clara, CA 95054
(408) 748-1875

D Y N A - M I T E v 4 . 0

Terminal Emulation Software with File Transfer for DOS Machines

The latest generation of the classic MITE datacomm package, just \$99.95!

- * XMODEM, XMODEM/Batch, YMODEM, Kermit, MITE and other Protocols
- * Duplicate "menu" and "command" style control mechanisms
- * Emulation of VT100, VT52, TV925, ADDS VP, ADM3a, Z19, IBM3101, others
- * MORSE programming language with variables, cond. branching, etc
- * Lower priced versions (with fewer features) available (incl. CP/M-80)

Mycroft Labs, Inc. / P.O. Box 4106 / Tallahassee FL 32315
Phone: 904 385-1141 Orders: 1 800 MYCROFT BBS: 904 385-MITE

object
\$189

BUSINESS BOARD SYSTEM

source
\$399

The Business Board System (BBS) provides a dial-up information center that can be accessed 24 hours a day via modem at 300/1200/2400 baud. The multi-level menu system is easily customized with no programming required. An unlimited number of data entry forms are user configurable and can be presented to callers based on their menu selections. The business applications are unlimited! This package is written in compiled BASIC and includes CONSOLX for remote PC operation and BORDM our data base Board Manager.

object
\$39

CASINO CARD GAMES

source
\$129

Treat yourself to these exciting video graphic card games just like you've seen in Vegas and Atlantic City! You get BlackJack, Slot Poker, and a "bluffing" game of Draw Poker! The package can run on either monochrome or color IBM PC/XT/AT & compatibles. It uses realistic card graphics and player "windows". BlackJack allows play variations and also displays statistics. Written in compiled BASIC. Basic Windows obj included with source.

object
\$49

BASIC WINDOWS

asm source
\$399

Create "instant" pull down menu's or pop-up help windows using any color combo's desired. Basic Windows uses a smart window buffer manager which does not use available BASIC memory! Also allows async operations for a dozen simultaneous windows. Screen save/restore, window create/restore/toss, window print using word wrap & other controls, Input string\$ from window, clear/scrollup/ scrollldn/clearEOL&EOS. Includes 2.5k code, doc, test program. For IBM & Compaq BASICA or IBM & MSQUICK compilers on Dos 2 or 3.



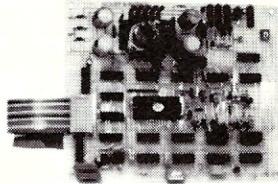
HAWKEYE
GRAFIX Inc

3415 HYDE PARK DRIVE
CLEARWATER, FL 33519
813-786-8161

Get our product brochure today
data encryption & time_billing
data comm & terminal emulation

CP/M, MS-DOS EPROM PROGRAMMING SYSTEM

2708
2758
2716
2516
2532*
2564*
68764*
2816A



2732/A
2764/A
27128/A
27256
27512
27CXXX
2864A
8751* MICRO C

- PAST ALGORITHM - STAND ALONE BOARD USES NO BACKPLANE SLOT
- NO PERSONALITY MODULES - INSTALL PROGRAM FOR SOFTWARE
- USES 24 VOLT XFRM FOR POWER - LARGE COMPREHENSIVE MANUAL
- PROGRAMS 262521 & 125V E/PROMS - ALL SUPPLIES ON BOARD
- NOT A SERIAL PROGRAMMER, NO FILE DOWNLOADING REQUIRED
- *SOCKET ADAPTER REQUIRED FOR DEVICES MARKED WITH ASTERISK(*) (ADAPTER DIAGRAMS INCLUDED)

PARALLEL PRINTER INTERFACE

CONNECTS TO ANY PARALLEL PRINTER INTERFACE - USES 8 OUTPUT DATA BITS AND ONE BIT FOR DATA INPUT (PRINTER BUSY LINE)

CONTROL PROGRAM COMMANDS

- PROGRAM EPROM(S) FROM DISK
- READ DISK FILE INTO RAM
- READ EPROM(S) INTO RAM
- VERIFY EPROM IS GRASBD
- DISPLAY/MODIFY RAM - (MONITOR MODE) WITH 14 SUB COMMANDS
- FILL-DUMP-XPER-EXAM-MODIFY-BIAS-PRGM-VERIFY-CKSUM, ETC)
- SAVE EPROM(S) TO DISK
- PROGRAM EPROM(S) FROM RAM
- COMPARE EPROM WITH RAM
- COPY EPROM

ASSEMBLED & TESTED UNIT WITH COMPLETE DOCUMENTATION AND SOFTWARE ON DISKETTE ---> \$199
PARTS KIT WITH SOFTWARE AND DOC->\$179 PCB SOFTWARE & DOC->\$69
SOFTWARE ON 8 & 5 1/4 DISK FOR KAYPRO, AMPRO, IBM & OTHER FMT'S

TO ORDER SEND CHECK, MONEY ORDER, WRITE OR CALL

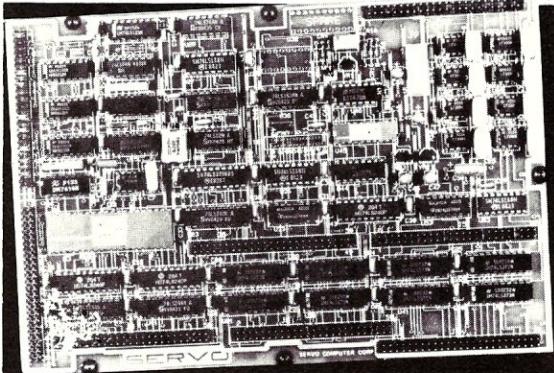
ANDRATECH
P.O. BOX 222
MILFORD, OHIO 45150
(513) 752-7218

CALL OR WRITE FOR MORE INFORMATION -- ADD \$4.00 FOR SHIPPING
OHIO RES. ADD 5.5% TAX -- VISA/M.C. ACCEPTED -- \$3.00 FOR COD

SERVO INDUSTRIAL PROCESS CONTROL BOARD

14 Bit Lab Accuracy Industrial Durability

The Features of 5 Boards in One.



The Servo Process Control Board is a general purpose multi-channel analog and digital input/output controller which can communicate with and control most medium-scale industrial processes. Designed to work with the Servo Industrial Microcomputer, (or other computers which support the SASI interface), it offers great accuracy at prices starting as low as \$495.

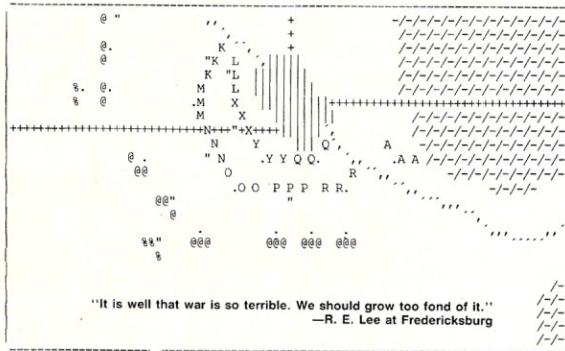
■ American made ■ MILSPEC components ■ 1 year warranty

Maximum configuration includes: ■ 192 analog inputs ■ 64 analog outputs ■ 128 digital inputs ■ 512 digital outputs

■ Based on the INTERSIL 14 Bit A to D converter ■ Communicates with any micro or mainframe computer ■ Easily programmed in high level languages (Turbo Pascal, Microsoft Basic, Forth) ■ PID, FFT, Thermo-couple linearization and laboratory DAS software supplied. ■ Custom software and hardware engineering assistance available.

SERVO
SERVO COMPUTER CORPORATION

360-B N. Ellensburg St.
Box 566
Gold Beach, OR. 97444
(503) 247-2021



Robert E. Lee would have been very fond of **GENERAL**. **GENERAL** simulates the anxiety, uncertainty, and despair felt by the commanders of armies in an imaginary war taking place a century or so ago. **GENERAL** is . . .

. . . a TWO-PLAYER game. It requires TWO CP/M-80 or IBM PC computers communicating via modem or cable at 300 baud. The computers are non-playing "umpires"; they produce statistical reports and animated displays of the battle situation, and relay the players' commands between the machines.

. . . a REAL-TIME game. It is always "your turn". The video display changes continually as time passes.

. . . a COMPLEX game. Each player controls an army of twenty-six units. There can be action in several places at once, and the decision which to attend to first can be critical.

. . . a SERIOUS game. It has had over five years of development and testing. A typical game lasts about two hours and requires constant, intense concentration. The game has an immense variety of possible situations, and does not grow stale with repeated playing.

GENERAL is available from:
Tee-Kay Software
P.O. Box 23771
Columbus, Ohio 43223

Diskette and User's Manual: \$50.00
User's Manual only: \$10.00
Serial I/O Software Listings: \$2.00

A variety of 5" CP/M disk formats is available (including Apple), as well as standard 8" SSSD. Specify desired format when ordering. NOTE: The CP/M version may require a small amount of customization of screen and modem control software. Sample ASM sources (8080 code) are included on the **GENERAL** diskette. Listings are available on request.

ISIS . . . CP/M MS-DOS

COMPLETE SOURCE CODE INCLUDED

ICXPDS: eXchanger now supports the 5 1/4" iPDS format. Manipulation of ISIS-II files using your computer system was never easier.

ICXMDS: Same as ICXPDS, but for MDS 8" systems.

IMXPDS: Reads/Writes 5" iPDS disks on PC's and AT's.

TELEDPLUS: Enhanced serial file transfer program for CP/M, ISIS, or MS-DOS.

ISE: Emulator gives the CP/M and MS-DOS user access to all the ISIS-II languages and utilities.

ACCEL 8/16: CP/M-80 emulator for MS-DOS. Enables PC's to run ISE. (no source code)

\$89 each
\$250 any 3

ZAS Development Package:

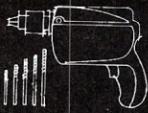
Z-8 / Z-8000 Assembler for CP/M and MS-DOS.

Request catalog of products



Western Wares 303-327-4898
Box C • Norwood, CO 81423

Copyrights: CP/M Digital Research, Inc.
ISIS-II and iPDS Intel Corp. - MSDOS Microsoft



Power Tools for system builders™

Call today for our free catalog of design aids, compilers, libraries, debuggers, and support tools for Apple and IBM micro computers. The Power Tools catalog includes product descriptions, warranty and license terms, and all the information you need to make an intelligent purchase decision.

TSF offers technical support, competitive pricing, free UPS shipping on orders over \$100, and a reasonable return policy. Visa, MasterCard, and American Express accepted without surcharge. **TSF helps you get your job done.**

Sample Prices:

Microsoft C \$259
MASM 4.0 \$109
Turbo Pascal \$45
Mark Williams C \$375
Lect C \$59
Wendin OS Toolbox \$89
Blaise Async Manager \$137

Call Toll Free
24 hrs a day/7 days a week

Ask For Operator 2053

800-543-6277

Calif: 800-368-7600



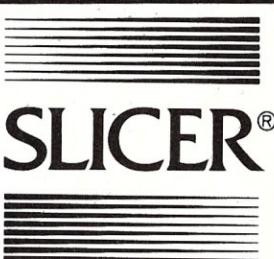
TSF
The Software Family™

- Dept. C-2 • 649 Mission Street
- San Francisco • CA 94105
- (415) 957-0111

THE PROVEN MICRO FAMILY for all the INDUSTRY APPLICATIONS

*Specials

New Low Prices!



*Multi-User Slicer Combo	\$950.00
Slicer SBC A&T (No RAM)		
Slicer 1 MEG Board A&T (Complete)		
CCP/M by Digital Research Inc.		
Slicer SBC A&T 256K	\$445.00
*Slicer SBC A&T 128K	\$350.00
with double deck sockets for additional 128K		
*Slicer SBC A&T (No RAM)	\$295.00
*Slicer 1 MEG Board A&T	\$450.00
*Slicer 1 MEG Board (Full Kit)	\$295.00
Slicer PC Expansion Board A&T	\$495.00
CCP/M (Digital Research Inc.)	\$250.00
MS DOS (Micro-Soft Corp.)	\$175.00
*New Slicer Bios for MS DOS	Call
Other kit forms available - Enclosures & other support hardware		
in stock - Call or write for latest information & prices!		

SLICER COMPUTERS INC.

2543 Marshall Street N.E.
Minneapolis, MN 55418

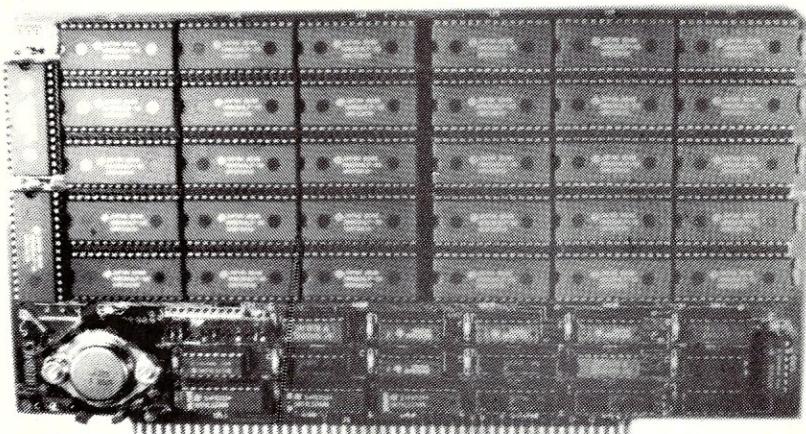
(612) 788-9481
BBS (612) 788-5909

256K BYTES OF 100ns STATIC MEMORY **\$395**

NO HIDDEN CHARGES!

Performics prices include dynamic burn in
and UPS 2nd day air delivery.

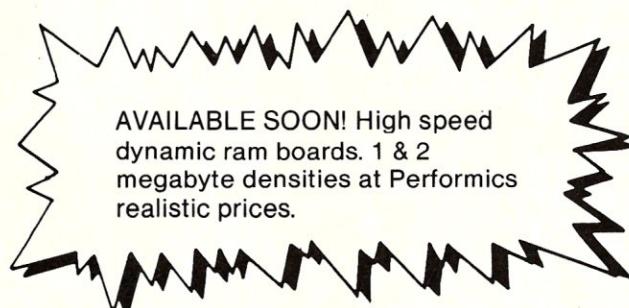
* 128K byte version (depopulated 256K) **\$345**



S-100

STANDARD FEATURES

- 100ns memory chips
- Operates in excess of 12 MHz
- Certified system ready
(industrial grade dynamic burn in)
- Supports 8 and 16 bit data transfers
- IEEE 696/S-100 compatible
- 24 bit addressing
- Address strappable to any 128K block within the 16 meg address range
- Extremely low power consumption
- Single +5 volt operation
- One year warranty



AVAILABLE SOON! High speed
dynamic ram boards. 1 & 2
megabyte densities at Performics
realistic prices.

PERFORMICS
INC.

Delivering high performance
products and excellent service
at realistic prices.

(603) 881-8334 ORDER NOW!

TERMS: Check, Visa/Mastercard

Performics guarantees our boards will work in your system or return
for full refund.

**11 Morning Dove Road • Kingston, NH 03848
• Hudson, NH 03051**



Interfacing Using The SCSI Bus

by Hui Tytus

Several years ago I installed a large capacity hard disk (65 Mbytes formatted) on a Radio Shack Model 2. Within a few months I had to switch to an S-100 setup using MS-DOS with an 8086. The Z80 and CPM-80 didn't provide the software or speed the application needed. Due to a lucky selection of components, the same disk, the controller and software, or nearly so, fit onto the new system with a minimum of trouble. Sound like magic? After all, those are two different architectures. Well, it is a bit of magic, and all due to the "Small Computer System Interface" (SCSI, pronounced "scuss-ee") bus with some help from the easily ported C language. The total cost, not counting some upgrading done at the same time, was \$200. Not bad.

The SCSI bus was first introduced with the acronym SASI by Shugart Associates to tie peripherals and one or more hosts together on small computer systems. It allows controllers to be mounted in the same external housing as the drives rather than on the host's bus. Consequently a new controller needn't be purchased for the new system. A host adapter, or personality board, is mounted on the host computer to tie in with all the devices on the SCSI bus. The result is similar to the effect of the RS-232 interface. When you switch your computer for a better model, the old printer and terminal are simply unplugged from the old and plugged into the new. With the SCSI bus, the same is possible with tape and disk drives *and* their controllers. Of course nothing involving computers is quite that simple, as a look at the following software driver proves. But the job is much simpler and much cheaper than working with proprietary interfaces.

A LITTLE SCSI & SASI HISTORY

SASI originally stood for Shugart Associates System Interface until an ANSI committee formalized the standard and gave it the name "Small Computer System Interface" (SCSI). The committee also added a few extra features that allow, say, a read sectors command to be started on one device disconnecting that device before completion, and then starting another task on another device, a tape drive for example. When the read sectors command is completed, that device can be

Using The SCSI Bus To Interface To A Hard Disk

reconnected to read the data from the controller's buffer.

The term SCSI has come to imply the ability to handle all features of the standard, including disconnect and reconnect, while the term SASI usually indicates only a subset of the commands are available. The scope of this article is limited to the subset of commands loosely defined by the term SASI.

The ANSI standard comprises two sections. The first is the SCSI bus hardware specification. The second is the protocol which defines the way commands, data, status, and messages are sent between the host and the controller, or, in SCSI lingo, the *initiator* and *target*. This protocol enables a driver, written in C (Listings 2 and 3), to be used with slight modification, on a wide variety of hosts. Between the portability of the C language and the portability of the SCSI protocol, the likelihood is that a driver need only be written once. The driver presented here was designed for a single user machine and is not interrupt driven. If I ever install a multi-tasking or multi-user system, an interrupt driven driver will have to be written. Maybe I should say there is hope it need be written only once.

INTERFACING TO THE BUS

The bus is simple, using a 50 pin ribbon cable to interconnect the disk drives, tape drives (if used), and the host(s). Eight leads are used for data with one parity line. Some controllers and host adapters utilize parity checking, others don't. Nine lines are used for control signals (select, busy, request, acknowledge, control/data, input/output, message, reset and attention). All control and data lines are biased at a positive voltage by terminating resistors and are considered to be asserted (logical 0) when pulled low to ground.

The sequence for the host to, for example, read a logical sector on a disk is divided into several phases. The selection phase requires the host to assert one of the data lines to identify which controller or *target* is desired and then assert the select line. The target addressed then asserts the busy line to indicate it has recognized its selection. As the SCSI idiom goes, it has been awakened.

In the command phase, the target unasserts the input/output line to indicate a block of six or ten bytes of command information should be outputted by the host to the target. The target asserts the request line for the first byte until the host asserts the acknowledge line to indicate the byte is on the data lines. When the target has read the byte, the request is unasserted. The host then unasserts the acknowledge line and the *handshake* sequence continues until all bytes of the command are transferred to the target.

Next, in the data phase, once the controller has read the sector(s) on the disk into a buffer, it will unassert the control/data line and assert the input/output line to indicate that the host can now input the data. This is done with the request/acknowledge handshake as before.

When all data is transferred, the status phase is entered and the controller asserts the control/data line to indicate the status byte may be input by the host. The status byte will indicate if all is well.

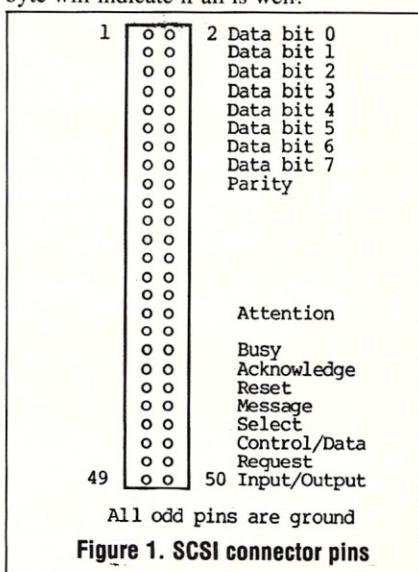


Figure 1. SCSI connector pins

Next, the message phase is indicated by the message line being asserted by the target. As a rule, this requires that the host read only one byte, which may be discarded.

There are several more phases provided by the SCSI standard, but those mentioned above are used in this driver and, for many controllers, they are the only ones available, i.e. the SASI specification.

A SCSI DEVICE DRIVER

The software shown in Listings 1 through 3 are the modules of an installable MS-DOS driver for a Priam 72 megabyte disk using an Adaptec 4000 or 4070 controller and a Data Technology Corp DTC 10-1 host adapter. Thanks to SCSI's hardware and software standards, the controller and host adapter can be replaced by units from other manufacturers with only minor modifications.

The first listing is the interface using the MS-DOS installable driver protocol and SCSI commands. It is written in 8086 assembly code and is compatible with the OBJ code produced by the Lattice C compiler. The third listing comprises the basic SCSI commands in C and is the principal object for any modifications that would be required for different components when using an MS-DOS operating system. If a different system is used, Listing 1 would have to be rewritten. Listing 2 could be largely deleted if only a basic driver is desired. Since it provides allocation of alternate sectors *on the fly*, it increases the effective reliability of the disk and simplifies the original formatting task at the same time. Without such a capability, the disk must first be formatted and then all sectors read to search for faulty sectors and a record kept. Then the disk must be reformatted with the location of the defective sectors furnished to the controller.

Before saying anything about Listing 1, it should be mentioned that the logical sector size used is 2K, which can cause trouble unless the system is booted from a drive with the same or larger size sectors. By altering the driver for the disk on which the system is booted, MS-DOS can be fooled into thinking that 2K sectors are used. This means some fiddling with another driver however. The simplest method I found was to patch MS-DOS with a new Maximum Sector Size Word (MSSW). That location on MS-DOS version 2.11 is 101h. Listing 4 shows some code within the operating system that will point to the MSSW. To find the MSSW on versions other than 2.11, I suggest searching for a portion of the code shown and then disassembling around that address in order to read the location of the MSSW. This has been done on version 2.10 effectively but not on any of the 3.xx versions. Also, note that version 2.10 has

Listing 1

```
; HDRV REV 2.11.03 850612 Hul Tytus
NAME C
extrn diskr:near
extrn init:near
extrn diskw:near
extrn maxblk:word
public in
public out
public drive
public dsgt

DRV_CNT equ 2 ; number of logical drives
TOTSECNT equ 2400
DGROUP GROUP DATA
PGROUP GROUP BASE,PROG
BASE SEGMENT WORD PUBLIC 'PROG'
ASSUME CS:PGROUP
ASSUME DS:DGROUP

        ; device header
        dw 0ffffh,0ffffh
        dw 0
        dw offset catch ; save add. to request header
        dw offset start ; start of driver code
unitc: db 2 ; number of units
        db 0,0,0,0,0,0
savess: dw 0
savesp: dw 0
reqhed: dw 0
        dw 0
bpb0: dw 800h ; bytes per sector
        db 1 ; sectors per cluster
        dw 1 ; reserved sectors
        db 2 ; number of fats
        dw 192 ; number of root dir entries
        ; 3 sectors
totsc0: dw TOTSECNT-33 ; total number of sectors
medis0: db 0f8h ; media descriptor
        dw 3 ; number of fat sectors
        dw 7 ; sectors per track
        dw 7 ; number of heads
        dw 0 ; number of hidden sectors
bpbl: dw 800h ; bytes per sector
        db 16 ; sectors per cluster
        dw 1 ; reserved sectors
        db 2 ; number of fats
        dw 192 ; number of root dir entries
        ; 3 sectors
totsc1: dw 0a5e0h ; total number of sectors
medis1: db 0f8h ; media descriptor
        dw 3 ; number of fat sectors
        dw 6 ; sectors per track
        dw 7 ; number of heads
        dw 0 ; number of hidden sectors
bpbpoint: dw offset bpbo
        dw offset bpbl
medpoint: dw offset medis0
        dw offset medis1
        db 'ERROR'


```

```
error: dw 0
jmptbl: dw offset hinit ; 0
        dw offset media ; 1
        dw offset bldpb ; 2
        dw offset dummy ; 3
        dw offset hread ; 4
        dw offset dummy ; 5
        dw offset dummy ; 6
        dw offset dummy ; 7
        dw offset hwrite ; 8
        dw offset hwrite ; 9
        dw offset dummy ; 10
        dw offset dummy ; 11
        dw offset dummy ; 12
; SCSI to MS DOS error code conversion
scsi: db 2,3,4,0dh,10h,11h,12h,13h,14h,15h,19h,20h,
        db 21h,22h,23h,24h,25h,98h,0
msdos: db 6,0ah,2,0,4,0bh,8,8,8,6,0ah,7,7,7,7,7
        db 7,4,0
C PROC FAR
catch: mov word ptr cs:[reqhed],bx ; store request header
        mov word ptr cs:[reqhed+2],es ; pointer
        ret
start: push ax ; start of driver
        push bx
        push cx
        push dx
        push bp
        push di
        push si
        push ds
        push es
        cli
        mov ax,ss ; save stack pointer
        mov word ptr cs:[savess],ax
        mov ax,sp
        mov word ptr cs:[savesp],ax
        mov ax,seg DGROUP
        mov bx,cs
        add ax,bx
        mov ds,ax
        mov ss,ax
        mov sp,offset DGROUP:SBASES
        sti
        mov si,word ptr cs:[reqhed] ; load request header
        mov es,word ptr cs:[reqhed+2] ; address
        xor ax,ax
        mov al,es:[si+1] ; remember which logical drive
        mov word ptr ds:[drive],ax
        mov cx,8101h
        cmp al,DRV_CNT ; make sure legal drive
        jnc lastxx
        mov ax,word ptr cs:[error] ; see if last call
        or ax,ax ; produced an error
        jz skpinit
        push si
        push es
        push ds
        pop es
        call init


```



**Don't be fooled.
No hidden cost!
Complete MS-DOS/CPM
Super Turbo**

In keeping with industry trends MSO is bringing our customers high performance P.C. compatibles and accessories. MSO takes the P.C. compatible to maximum performance with its SUPER TURBO. The SUPER TURBO features the V20-8 chip which runs at three times the speed of the IBM-PC XT* and also runs CPM 8080 software.

The SUPER TURBO comes complete with the MS-DOS operating system, Read and Run CPM, full Instructional Documentation, Utility software, **plus for our first 100 customers MSO is offering the Micropro Wordstar Professional Software package.** This package includes: Wordstar, Mailmerge, Correctstar, Starindex, Datastar and a G.L. Accounting System. The SUPER TURBO is a complete turnkey system with everything necessary to plug in and operate.

**USER SUPPORT
HOT LINE**

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

For our catalog with complete details and prices, send \$2.00 to:

Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

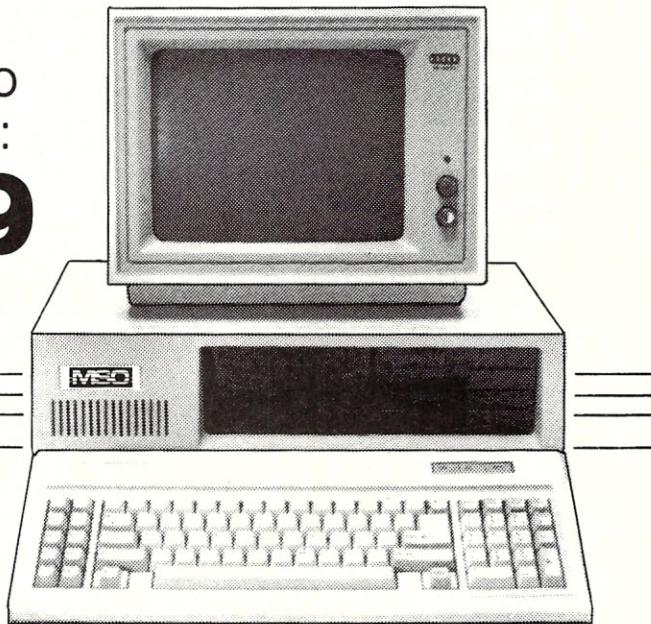
MON. - FRI. 7am - 5pm PST • SAT. 9am - 5pm PST

*IBM is a registered trademark of the IBM Corporation.

15% restocking on returned orders

dare to compare.

Super Turbo
Super Price:
\$1299



The Super Turbo P.C. runs IBM software and CPM 8080 programs

- CPU - V20-8 8mHz Super Chip runs *IBM compatible software at 3 times the speed of the IBM-XT and CP/M 8080 software.
- 8087 Math Processor optional
- 256K RAM on mother board expandable to 640K
- ROM 8K Bios
- 6 empty slots for expansion
- 2 serial port one optional with expansion kit
- 1 parallel port
- 1 game port
- Clock calendar with software
- Hi-Res monographics video board
- Floppy controller
- Dual Floppy Drives 360K ea.
- 135 watt XT Power Supply
- 5150 style compatible keyboard
- Hi-Res TTL Green or Amber 12" monitor
- MS-DOS operating system and manual.
- Instructional Documentation and Utility Software
- Assembled and tested in U.S.A.
- **Optional internal 20 meg sub system for Super Turbo add \$499**

Special printer pricing with purchase of above computer.



MSO, the leader in cost effective computer systems, is offering the Sanyo MBC 1200 at a **giveaway price!**

The MBC 1200 Accounting Software
-G/L, A/R, A/P, Inventory and Payroll modules retail for \$398 per module. Now from MSO you pay only \$99 per module or all five for \$299! Second disk drive with fan-option available. COMPAT disk utility available.

SPECIFICATIONS:

- Two Z-80A (main and subsidiary) CPUs with no-wait mode for fast execution, substantial memory capacity (RAM 64KB, ROM 4KB).
- High-resolution full graphic function with 640 x 400 dot matrix display.
- Choice of 33 or 40 line text mode.
- CP/M operating system complete with interpreter, editor and all utilities.
- Easy-to-use Sanyo graphic BASIC.
- One (MBC 1200) or two (MBC 1250) internal double-sided, double-density, double-track, 5 1/4" slim-type 640KB formatted mini floppy disk drives.
- Special design featuring detachable ergonomic keyboard with coiled cable.
- Interfaces for Centronics printer and one RS-232C port provided.

20 meg internal hard drive subsystem for Sanyo MBC 1100, 1160 & 1200

\$599 (with purchase)

Printer specials with purchase

805/393-2247

All systems carry full 90 day warranty.

CASH PRICE ONLY

Check in advance. Add 3% for VISA/MC. Shipping & handling charges will be added to each order.

For our catalog with complete details and prices, send \$2 to:

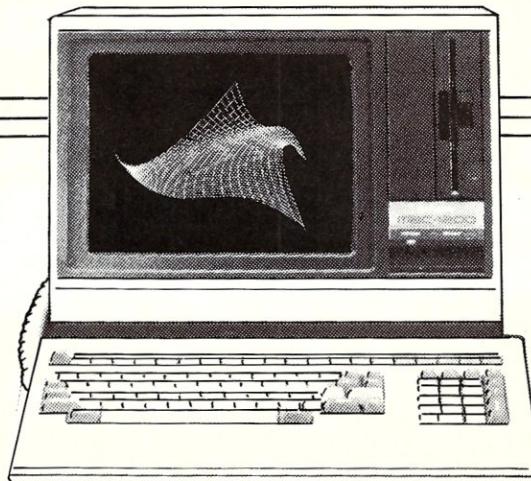
Micro Supply Organization, Inc.
4909 Stockdale Hwy. #180
Bakersfield, CA 93309

15% Restocking on Returned Orders



the giveaway.

SANYO
MBC 1200



FREE SOFTWARE

- Wordstar • Spellstar
- Calcstar • Mailmerge
- Infostar • Basic

\$399

PC/VI

Full Screen Editor for MS-DOS (PC-DOS)

Looking for an Ultra-Powerful Full-Screen editor for your MS-DOS or PC-DOS system? Are you looking for an editor FULLY COMPATIBLE with the UNIX* VI editor. Are you looking for an editor which not only runs on IBM-PC's and compatibles, but ANY MS-DOS system? Are you looking for an editor which provides power and flexibility for both programming and text editing? If you are, then look no further because PC/VI IS HERE!

The following is only a hint of the power behind PC/VI: English-like syntax in command mode; mnemonic control sequences in visual mode; full undo capability; deletions, changes and cursor positioning on character, word, line, sentence, paragraph or global basis; editing of files larger than available memory; powerful pattern matching capability for searches and substitutions; ability to invoke other programs from within PC/VI; macro capability; location marking; joining multiple lines; auto-indentation; word abbreviations and MUCH, MUCH MORE!

The PC/VI editor is available for IBM-PC's and generic MS-DOS based systems for only \$149. Available in 5 1/4", 8" and 3 1/2" disk formats. Minimal System Requirements: MS-DOS Version 2.00, 256KB, 1 disk drive. Recommended system: 512KB and hard disk. Quantity and dealer discounts available. For more information call or write:

Custom Software Systems
P.O. Box 551 MO
Shrewsbury, MA 01545
617-842-1712

The UNIX community has been using the VI editor for years. Now you can run an implementation of the same editor under MS-DOS. Don't miss out on the power of PC/VI!

*UNIX is a trademark of AT&T Bell Laboratories.

a bug that causes the buffer area to be short 16 bytes with sectors larger than 512 bytes and would probably not work correctly with the larger sectors. Keep in mind that the file MSDOS.SYS or IBMDOS.SYS must be loaded into RAM fresh from the disk. The search cannot be made once MS-DOS becomes resident because the code shown is discarded after it is used to configure the buffers.

The assembly code provides MS-DOS with two BPB (BIOS Parameter Block) fields containing the essential information about the drive. Two are used because the drive is split into two logical partitions. The first logical drive, or partition, is about 5 megabytes and the second is equal to the remainder. Only 5 MS-DOS commands are handled by this code.

The first is the MS-DOS INIT command which is channeled through the jump table *jmpltbl* to the routine *hinit*. The BUILD BPB command is channeled to the routine *bldbpb* and the MEDIA CHECK command calls *media*. The read and write commands are handled by *hread* and *hwrite* respectively. The comments in the listing provide some insight, but to implement this driver on a system, I recommend purchasing Microsoft's *Programmer's Reference*. It is often available from OEMs who use MS-DOS. IBM has a version that includes PC-DOS particulars, but, at more than \$100, it is awfully expensive. Radio Shack has a version at around \$15.

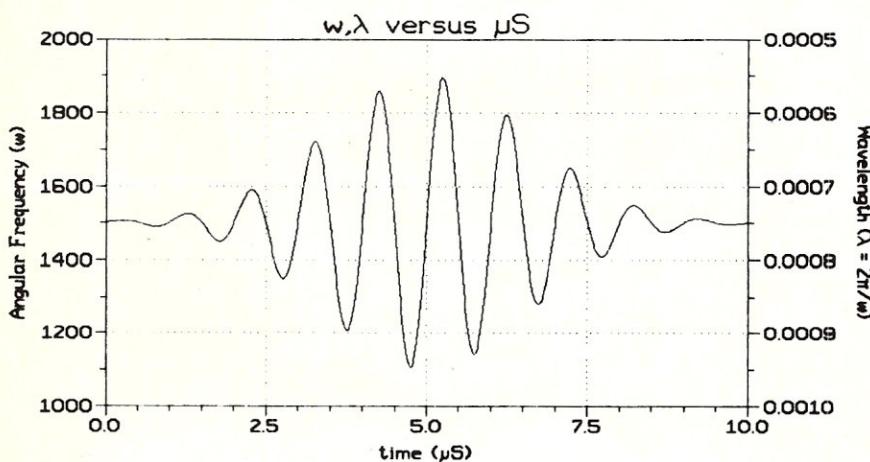
For some specific detail, the size of the first drive is held in the "totsct0" field of *bpb0*. This defines the number of logical sectors and can be any value less than the total available on the drive itself. The driver can be changed to provide only one logical drive by deleting the line after *start*: that reads "mov word ptr ds:[drive],ax", setting DRVCNT to 1, and initializing the field *drive* at 1. Also, the array *drvbg[]* in Listing 2 should be initialized as { 0, 0 }.

The Winchester disks this driver is designed to control are much faster and more reliable than floppies. The precision manufacture of hard drives allows the disk to be sealed within an ultra clean environment and the read-write heads ride slightly above the media rather than in contact with the disk accounting for the reliability. Also, most hard disk controllers provide error correcting circuitry (ECC) that allow burst errors of up to 15 bits to be recognized and corrected. Eight bit burst correction is provided on the Adaptec controllers.

Even with such safeguards, disks can be counted upon to have some unusable sectors, usually due to imperfections in the media. These drives are typically shipped from the manufacturer with a list of faults and their locations. The various SCSI controllers all have some form of firmware allowing these fault locations to be remembered and either skipped or sub-continued on page 53.

GRAPHICS

starting at \$49⁹⁵



SCI-GRAF: Produces line, scatter, and high/low plots on Epson or IBM compatible dot-matrix graphics printers. Supports images up to 1680x1712 dots (over 3 million pixels!), log & linear scales, graphs in 7 colors (on a JX-80), batch processing. Requires MS-DOS 2 or 3, 256K ... \$99⁹⁵

FONTEdit: Easy-to-use full screen font editor allows you to create and modify fonts for our SCI-GRAF program. Create Greek, math, and custom symbols in 3 sizes. Supports SCI-GRAF's proportional spacing. Requires IBM compatible 320x200 pixel on-screen graphics & IBM compatible keyboard \$39⁹⁵

GRAF 3.0: Produces bar, pie, line, scatter, and high/low plots on Epson, IBM, C.Itoh, or NEC compatible dot-matrix graphics printers. Supports 14 fill-in patterns, 8 point-plotting symbols, automatic scaling, labeling, and legend creation. MS-DOS 2 or 3, 192K \$69⁹⁵
CP/M-80: Requires 64K (54k TPA) \$49⁹⁵

MSC Microcomputer
Systems
Consultants

27 Forest Avenue Port Jefferson Station New York 11776-1820 (516) 928-7493

```

pop es
pop si
call setcnt
skpinit:
    mov al,es:[si+2]
    mov cx,8103h
    cmp al,0ch           ; check for legal command
    jnc lastxx
    shl ax,1
    mov bx,offset cs:jmptbl
    add bx,ax
    call cs:[bx]          ; call command
    mov word ptr cs:[error],ax ; save error code
    or al,al
    mov cx,100h
    jz lastxx
    call seterr
lastxx:
    mov word ptr es:[si +3],cx ; load request header
    mov word ptr cs:[savess] ; with status code
    cli
    mov ax,word ptr cs:[savess] ; reload stack pointer
    mov ss,ax
    mov ax,word ptr cs:[savesp]
    mov sp,ax
    sti
    pop es
    pop ds
    pop si
    pop di
    pop bp
    pop dx
    pop cx
    pop bx
    pop ax
    ret
; Come here to abort
;ABORT:
    mov cx,810ch
    jmp lastxx
C
ENDP

hinit:
    push si
    push es
    push ds
    pop es
    call init           ; initialize
    pop es
    pop si
    push ax
    mov al,byte ptr cs:[unitc] ; load req head with # of
    mov byte ptr es:[si+0dh],al ; logical drives
    mov bx,offset cs:bpbpoint ; get pointer to BPB pointer tbl
    mov bx,word ptr ds:[drive] ; figure which BPB tbl
    shl bx,1
    add ax,bx
    mov word ptr es:[si+12h],ax ; load req header with pointer
    mov ax,cs            ; to BPB pointer
    mov word ptr es:[si+14h],ax
    xor ax,ax
    mov word ptr es:[si+0eh],ax ; load req head with address of
    mov ax,seg buffer      ; end of driver
    mov bx,cs
    add ax,bx
    mov word ptr es:[si+10h],ax
    pop ax
    push ax
    call setcnt          ; set # of logical sectors
    ; print install notice only once
    mov al,byte ptr cs:[first]
    or al,al
    jnz hinitl
    inc al
    mov byte ptr cs:[first],al
    push ds
    push es
    push si
    push cs
    pop ds
    mov dx,offset signon
    mov ah,9
    int 21h
    pop si
    pop es
    pop ds
hinitl:
    xor ax,ax
    pop ax
    ret
setcnt:           ; sets number of sectors in
                    ; logical drive 2
                    ; ax--> if not 0, will abort
    push ax
    push bx
push cx
push dx
mov dx,ax
mov ax,word ptr [maxblk] ; get # of physical sectors
mov bx,word ptr [maxblk+2]
mov cx,2
setcntl:           ; change to logical sectors
    shr bx,1
    rcr ax,1
    loop setcntl
    or dx,dx
    jnz setcntrfin
    mov bx,TOTSECTN+33
    sub ax,bx           ; sub logical sect's in drive 1
    and ax,0fff0h ; make sure there is no partial cluster
    mov word ptr cs:[totscnt],ax
setcntrfin:
    pop dx
    pop cx
    pop bx
    pop ax
    ret
first: db 0
signon: db 'Hd.sys version 2.11.03 installed.',0dh,0ah,'$'
media:
    mov ax,word ptr cs:[error]
    or ax,ax
    mov al,0ffh           ; if error last time
    jnz medial           ; set media changed status
    mov bx,word ptr ds:[drive]
    shl bx,1
    mov ax,offset medpoint
    add bx,ax
    mov bx,word ptr cs:[bx]
    mov al,byte ptr cs:[bx]
    mov bl,es:[si+0dh]
    cmp al,bl           ; if MSDOS matches driver's media byte
    mov al,1            ; set no change status
    jz medial
    mov al,0ffh           ; else media changed status
medial:
    mov byte ptr es:[si+0eh],al ;load req head with media status
    xor ax,ax
    ret
bldbpb:
    mov bx,offset cs:bpbpoint ; load req head with pointer to
    mov ax,word ptr ds:[drive] ; correct BPB table
    shl ax,1
    add bx,ax
    mov ax,word ptr cs:[bx]
    mov word ptr es:[si+12h],ax
    mov ax,cs
    mov word ptr es:[si+14h],ax
    xor ax,ax
    ret
hread:
    push es           ; pass arguments to diskr in lst #2
    push si
    mov bp,sp
    mov ax,word ptr es:[si + 12h]
    push ax           ; count
    mov ax,word ptr es:[si + 10h]
    push ax           ; transfer seg
    mov ax,word ptr es:[si+0eh]
    push ax           ; transfer byte
    xor ax,ax
    push ax           ; sector MSW
    mov ax,word ptr es:[si+14h]
    push ax           ; sector LSW
    push ds
    pop es
    call diskr
    mov sp,bp
    pop si
    pop es
    or ax,ax
    jz hreadl
    xor cx,cx       ; if error, tell MSDOS 0 sect's read
    mov word ptr es:[si+12h],cx
hreadl:
    ret
hwrite:
    push es           ; pass arguments to diskw
    push si
    mov bp,sp
    mov ax,word ptr es:[si + 12h]
    push ax           ; count
    mov ax,word ptr es:[si + 10h]
    push ax           ; transfer seg
    mov ax,word ptr es:[si+0eh]
    push ax           ; transfer byte
    xor ax,ax
    push ax           ; sector MSW
    mov ax,word ptr es:[si+14h]
    push ax           ; sector LSW
    push ds
    pop es
    call diskw

```

RP/M2™ creates CP/M®2.2 compatible IBM PC

1. Remove the 8088
2. Install the NEC µPD70108 (V20)
3. Boot PC RP/M2

The µPD70108 chip is a fast 8088 that also executes 8080 machine code. PC RP/M2 is an **operating system**. Either standalone or with MSDOS present, PC RP/M2 provides the solid base of a genuine operating system reliably distinct from the facade created by an MSDOS interface. All 2.2 features are supported with 56.5k TPA, plus file date and time stamping, fast virtual disk, iobyte redirection, terminal cursor positioning, color console display. DOSDISK accesses MSDOS drives; MODEMPC transfers files via serial port; SETDISK redefines a drive to any of over 80 formats, allowing PC RP/M2 to directly process existing CP/M files in their native form. Two system disks plus 105 page user and technical manual \$129. Shipping \$5 (\$10 nonUS)  

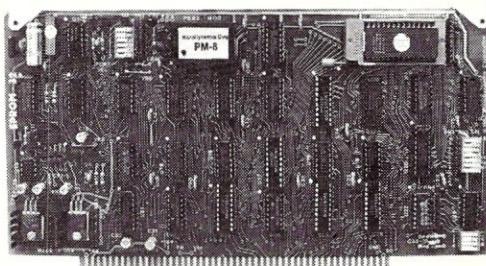


118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

Call for Update

S-100 EPROM PROGRAMMER

EPROM-32



- Field-proven board meets IEEE-696 standard.
- Programs 1K through 32K (byte) EPROMs.
- Textool zero-insertion-force programming socket.
- EPROM is programmed through I/O ports and can be verified through I/O ports or located in memory space for verification.
- Programming voltage generated on-board.
- Personality Modules adapt board to EPROMs:

PM-1—2508, 2758	PM-3—2732, 2732A	PM-6—68764
2516, 2716	PM-4—2564	PM-8—27128
PM-2—2532	PM-5—2764	PM-9—27256

- Feature-packed CP/M-compatible control software includes fast programming algorithm.
- One year warranty.

\$269.95*
(A & T)

MicroDynamics

Corporation

P.O. Box 17577 • Memphis, TN 38187
(901)-755-0619

Price includes EPROM-32, documentation and two Personality Modules (specify). Additional Modules—\$17.95. Control software on 8" SSSD diskette—\$29.95. UPS ground—\$2.00, UPS air—\$4.00, COD—\$2.00, foreign add \$15.00. VISA & MASTERCARD welcome.

See Dec. 1983 **Microsystems** for a review of the EPROM-32.

```

mov sp,bp
pop si
pop es
or ax,ax
jz hreadl
xor cx,cx ; if error, tell MSDOS 0 sect's written
mov word ptr es:[si+l2h],cx
hwrite:
ret
dummy: ; just in case MSDOS sends unused command
xor ax,ax
ret
seterr: ; al = error in, cx=error out
mov bx,offset scsi ; load error code table pointers
mov dx,offset msdos
mov cx,810ch ; load default error
setrep:
mov ah,byte ptr cs:[bx]
or ah,ah ; if end of scsi table,
jnz seterr1
ret ; return w/ default error
seterr1:
cmp ah,al
jz seterr2
inc bx
inc dx
jmp setrep ; try again
seterr2:
mov bx,dx ; since SCSI match was found,
mov cl,byte ptr cs:[bx] ; return MSDOS equivalent
ret
in proc near
pop bx
pop dx
push dx
push bx
xor ax,ax
in al,dx
ret
in endp
out proc near
pop bx
pop dx
pop ax
push ax

```

```

push dx
push bx
out dx,al
ret
out endp
dsget proc near
mov ax,ds
ret
dsget endp
BASE ENDS
STKRSV EQU 256 ; reserved stack size
DATA SEGMENT PARA PUBLIC 'DATA'
PUBLIC BASE
SBASE DB STKRSV DUP (?)
SEASSES DB 0,0
BASE DW OFFSET DGROUP:SBASE
NULL DW 0
drive DW 0
DATA ENDS
PROG SEGMENT BYTE PUBLIC 'PROG'
ASSUME CS:PGROUP
ASSUME DS:DGROUP
nop
nop
PROG ENDS
buffer segment memory 'last'
db 0,0
buffer ends
END

```

stituted with alternates. Unfortunately, first ascertaining and then transferring the defect list to the controller is a cumbersome operation which complicates the original formatting of the disk.

This would not pose a problem if a disk could be relied upon to never develop unusable sectors, after formatting. Such perfection, however, is not possible. Listing 2 handles a fault occurring in the middle of an operation without causing a halt in the operation. In other words, degraded sectors are given alternates *on the fly*.

127 physical sectors are set aside as alternates. Before a read or write operation (diskw() or diskr()) is started, the function trans() is called to provide the correct address and make sure that all the sectors required in the operation are in their *natural* location. If not, that is, if one or more is defective and has been assigned an alternate, each physical sector is read or written individually by the function dfix(). Dfix() is also used if a new defective sector is discovered.

If, in a read operation, the error correction code must be used, an alternate sector will be applied even if the correction is successful. The idea is to keep error correction in reserve to be used only as a last resort. If the sector is bad, avoid it.

Two arrays of addresses are maintained, both on the disk and in RAM. The first array contains the natural addresses of unusable sectors and the second holds the locations of the alternates. When a sector is assigned an alternate, the arrays in RAM are set to provide the proper addressing and then written to the disk. Two copies are made for safety. The string "BADBLK" directly precedes the arrays in RAM so that Debug can be used to first find the string, and then display the data held in the arrays. This provides a simple means of viewing the number of unusable sectors. Since many controllers provide alternate sector capability at format time, if the sectors are known, this will provide the necessary information.

Listing 3 provides the basic I/O to the SCSI bus. There are two types of functions: first the phase routines that execute one of the basic steps, or phases, in the SCSI protocol and, second, the commands, like readc(), that execute one of the commands defined by the protocol. The second are just distinct combinations of the phase instructions. Due to the standardized interface, this works on most controllers with the exception of the dwrite() and dread() functions that use the proprietary DMA capability of the DTC 10-1.

Getcon() handles the selection phase by first checking the busy line to be sure there is no activity on the bus. Then it sets the appropriate bit on the data bus to identify the target, or addressed controller.

Listing 2

```
/* hdctrl      rev 2.11.03          850612  Hul Tytus */
#define READ 2           /* read command for dfix() */
#define WRITE 1          /* write command for dfix() */
#define NOSPC 0xd        /* out of space for alternate block */
#define SOFTER 0xf       /* software error in the driver */
#define DNR 4             /* drive not ready */
#define BADADD 0x21       /* illegal block address */
#define SEEK 2            /* drive seek error */
#define CSEEK 0x15        /* controller seek error */
#define ECCERR 0x98       /* error code when ECC must be used */
#define ALT_CNT 127       /* number of bad blocks permissible */
#define ALT_TOT 128       /* ALT_CNT + 1 must be binary unit */
#define BIN_CNT 64         /* = ALT_TOT/2 */
#define ALTEXP 7           /* ALT_TOT = 2**ALTEXP */
#define ARYBLK 3           /* start of blocks for the arrays */
#define ALTBLK 7           /* first alternate block */
#define BLKSIZ 512         /* size of blocks in bytes */
#define BLKINC 0x200000L    /* BLKINC = BLKSIZ << 12 */
#define BADCNT 2           /* number of blocks to hold both arrays */
#define FSTBLK 134         /* remember these are duplicated */
#define SHIFT 2             /* # of left shifts so that */
#define CYLSIZ 190         /* # 512 byte blocks = count << SHIFT */
extern unsigned drive; /* determines which offset to use */
/* if only one logical drive, make this 0 */
static char bloop[6] = {'B', 'A', 'D', 'B', 'L', 'K'};
static long badblk[ALT_TOT]; /* table of bad blocks */
static long newadd[ALT_TOT]; /* table of replacement blocks */
long lstblk = 0;           /* last usable block */
/* you should use readc for this */
long maxblk = 0;           /* last system block */
/* should be lstblk -134 in init after readc */
static int altnflg;        /* number for array or zero */
static int okcount;         /* = 1 if all sectors are natural */
                           /* start of logical drive offset in 512 sectors */
long drvbg[2] = {0, 9600};
long readc();
long trans();
int diskw(block, point, count) /* write count sectors starting at block */
                               /* from point(er) in memory */
long block, point;
int count;
{
    long sect;
    int x;
    block = block << SHIFT; /* convert logical 2048 byte sectors */
    count = count << SHIFT; /* to physical 512 byte sectors */
    block = block + drvbg[drive]; /* add logical drive offset */
    sect = trans(block, count);
    if (okcount) /* if an alternate sector is not used */
    {
        x = dwrite(sect, point, count);
        if (!x)
            return 0;
    }
    while (count--) /* if error or alternate sector is required */
    {
        x = dfix(block, point, WRITE);
        if (x)
            return x;
        point = point + BLKINC; /* increment pointer & block */
        ++block;
    }
    return 0;
}

int diskr(block, point, count) /* read count sectors to point */
long block, point;
int count;
{
    long sect;
    int x;
    block = block << SHIFT; /* convert logical 2048 byte sectors */
    count = count << SHIFT; /* to physical 512 byte sectors */
    block = block + drvbg[drive]; /* add logical drive offset */
    sect = trans(block, count);
    if (okcount) /* if an alternate sector is not used */
    {
        x = dread(sect, point, count);
        if (!x)
            return 0;
    }
    while (count--) /* if error or alternate sector is required */
    {
        x = dfix(block, point, READ);
        if (x)
            return x;
        point = point + BLKINC; /* increment pointer & block */
        ++block;
    }
    return 0;
}
```

```

int dfix(blockk, src, cmd) /* cmd = 1 for write, 2 for read */
/* this is routine that handles read & write errors */
long blockk, src;
int cmd;
{
    long block, check;
    int x, y, z;
start:
    y = 2;
    while(y--) { /* try y times to get it right */
        block = trans(blockk, 1); /* get correct block number */
        if (!block)
            return BADADD; /* 0 return means illegal block number */
        if (cmd == WRITE)
            x = dwrite(block, src, 1);
        else x = dread(block, src, 1);

        if (!x) /* if no error, return no error */
            return 0;
        if (x == DNR) /* if drive not ready, return */
            return x;
        /* figure a block in next cylinder */
        check = block - CYLSIZ;
        if (check < 0)
            check = block + CYLSIZ;
        vfy(check, 1); /* wiggle head */
    }

    z = x;
    /* retries didn't work, so set alternate block */
    /* if correctable error */
    if ((x > 0xF && x < 0x20 && cmd == WRITE) || x == ECCERR)
    {
        x = place(block); /* assign alternate block */
        if (x)
            return x; /* return if error */
        putary(); /* write RAM alternate tables to disk */
        /* if ECC error on read, write alt block w/ corrected data */
        if (z == ECCERR && cmd == READ)
        {
            block = trans(blockk, 1);
            x = dwrite(block, src, 1);
            if (!x)
                return 0;
        }
        goto start; /* now try again with alternate block */
    }
    return x;
}

long trans(block, cnt) /* translates logical to physical sector */
/* returns 0 for error */
/* okcount = 1 if all sectors are natural */
int cnt;
long block;
{
    int x, y;
    block = block + FSTBLK;
    if (block > maxblk)
    {
        altflg = 0;
        return 0;
    }
    x = BINCNT; /* x = mid point of possible alternates */
    y = badblk[0]; /* y = number of actual alternates */
    while (x > y) /* divide x by 2 until x becomes binary */
        x = x >> 1; /* mid point of actual alternates */
    altflg = x;
    okcount = 0;

    while (x) /* do binary search to see if block has an alternate */
    {
        x = x >> 1;
        if (block == badblk[altnflg]) /* if block has an alternate */
            return (newadd[altnflg]); /* return alternate */
        if (block > badblk[altnflg] && badblk[altnflg] > 0)
            altnflg = altnflg + x;
        else altnflg = altnflg - x;
    }

    if (block > badblk[altnflg])
        if (altnflg == ALICNT)
            okcount = 1;
        else ++altnflg;
    /* now check to see if any of next blocks have alternates */
    if ((badblk[altnflg]) >= (block + (long) cnt))
        okcount = 1;
    /* indicates that all blocks in transfer are not alternates */
    if (!badblk[altnflg])
        okcount = 1;
    altnflg = 0;
    return block;
}

int place(blockk) /* if altnflg=0, inserts new block to badblk[] & */
/* newadd[] and changes altnflg */
/* if altnflg>0, inserts new # to newadd[altnflg] */

```

When the selection line is asserted, the target acknowledges its ID number by asserting the busy line. A timeout is provided in case the controller is not present. The selection phase ends by the host unasserting the select line.

The request() function waits for the request line to be asserted by the target. As described previously, this is the start of the handshake for transmission between host, or initiator, and the target.

Putcmd() transmits the six or ten byte command string to the controller to complete the command phase. Each byte is not counted, rather the routine is allowed to send the command bytes until the control lines change to indicate a new phase should be entered. This allows the controller to abort at any time without befuddling the host.

Putdat() and getdat() take care of the data phase. These routines are used when the command requires more data than is included in the proper command string. Notice these do not count the number of bytes but continue to transmit until the controller indicates otherwise.

Status() does just what the name suggests; it returns the status or check byte. Along with providing the status phase, the message byte is also read and discarded. The message phase is not used in this driver except to terminate an operation.

Sense() is actually a complete command that uses all of the phases mentioned to sense the condition of the controller and drive. This is necessary because the check byte only indicates if an abnormality has occurred. Sense() returns the specific error code.

The remaining functions use those basic functions already mentioned to fully execute the ANSI standard commands. There are two exceptions. Both dread() and dwrite() use the dmaio() function which is special to the DTC 10-1 DMA circuitry. For other host adapters with no DMA, putdat() and getdat() could be mimicked in assembly code to handle the full address range required by the system.

After about six months in service, this driver has proved to be quite adequate. Unfortunately, in a backwards way of speaking, the 5" Priam disks have also proved to be quite reliable. This means that the drivers have not had a truly exhaustive test. Such a situation was anticipated in the design considerations leading to the belief that a simple scheme for allocating alternate sectors would not impair the speed of the disks in any significant way. Such is the case.

A small C program, written to check the read and write throughput, indicates about 300K bytes per second for reading and about half that for writing, using Adaptec's 4070 controller. The reduction continued on page 67

```

long blockk;
{
    int x, y;
    long block;
    block = blockk;
    /* check to see if an alt block is available */
    if (newadd[0] >= ALTCNT)
        return NOSPC;
    if (altnflg)           /* block is already marked bad */
        goto lump;
    ++badblk[0];          /* increment length of array */
    x = 1;                /* and find location for block */
    while (block > badblk[x] && badblk[x] != 0 && x < ALTCNT)
        ++x;
    y = badblk[0]; /* y = last element in array when finished */
    /* move upper part of array up one element */
    while (y > x)
    {
        badblk[y] = badblk[y - 1];
        newadd[y] = newadd[y - 1];
        --y;
    }
    badblk[x] = block;      /* insert new bad block */
    altnflg = x;            /* block is now marked */
lump:
    x = newblk(altnflg);   /* get new alternate sector */
    return x;
}

int newblk(arynum)      /* gives new block # to newadd[altnflg] */
/* returns error if all alternates in use */
int arynum;
{
    if (newadd[0] >= ALTCNT) /* return error if table is full */
        return NOSPC;
    newadd[arynum] = newadd[0] + ALTBLOCK;
    ++newadd[0];           /* increment total number of alternates */
    return 0;
}

int putary()           /* puts both arrays to the disk */
/* remember to declare badblk[] just before newadd[] */
{
    int x, y, ds;
    long blk;
    char *sourc;
    ds = dsget(); /* get current ds register */
    blk = ARYBLK;
    x = 2;           /* remember badblks are duplicated */
    while (x--)
    {
        sourc = &badblk[0];
        y = BADCNT;
        while (y--)
        {
            /* sourc & ds simulate a long that holds segment & pointer registers */
            dwrite(blk, sourc, ds, 1);
            sourc = sourc + BLKSIZ;
            ++blk;
        }
    }
}

int getary()           /* moves array from disk to ram */
{
    char *dest;
    int y, x, ds;
    long block;
    ds=dsget();           /* get current ds register */
    dest = &badblk[0];
    block = ARYBLK;
    y = BADCNT;
    while (y--)
    {
        /* dest & ds simulate a 32 bit pointer */
        x = dread(block, dest, ds, 1);
        if (x) /* if error, get from second disk copy of tables */
        {
            x = dread((block + BADCNT), dest, ds, 1);
            if (x && x != ECCERR)
                return x;
        }
        dest = dest + BLKSIZ;
        ++block;
    }
    return 0;
}

int init()             /* initialize driver and controller */
{
    int x;
    x=reinit();
    if(x) return x;
    lstblk = readc();
    maxblk = lstblk - FSTBLK;
    if (lstblk < 256) /* if error is reported */
    {
        x = lstblk;
    }
}

```

DeSmet C

— now with —

LARGE MEMORY OPTION

C88 Compiler \$109

Full K&R + V7 extensions
inline asm
assembler, linker, librarian
full screen editor (SEEtm),
8087 & S/W floating point

D88 Debugger \$50

Set Breakpoints by line number
or function name
Examine Global/Local variables
by name
Show C source while debugging
Both D88 and User displays

Large Memory \$50 Model Option

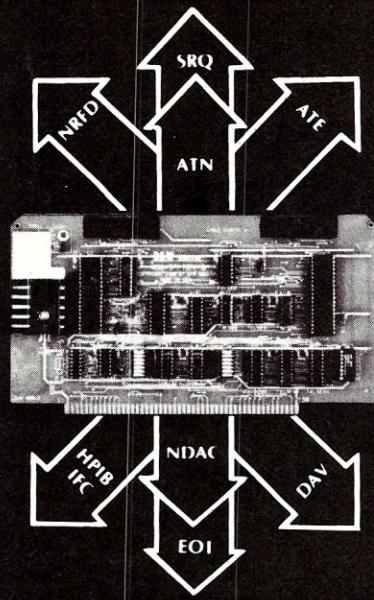
32-Bit Pointers (Full Megabyte
Addressing)
Fast Access of Static Data
Works with D88

— plus —

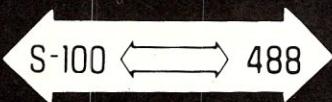
DOS Link	\$35
convert to .OBJ files	
Graphics (+src)	\$35
Hacker	\$25
source for start-up, RAM.COM & more	
Make	\$50
full UNIX-level	
Tools (+source)	\$35
XARRAY	\$39
Large Arrays (+source)	

C WARE CORPORATION

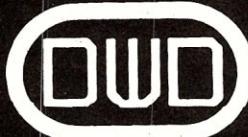
505 W. OLIVE, SUITE 767
SUNNYVALE, CA 94086 USA
(408) 720-9696 TELEX: 358185
We accept
VISA, MC & American Express



THE 488+3 IEEE 488 TO S-100 INTERFACE



- Controls IEEE 488 (HP1B) Instruments with an S-100 computer
- Acts as controller or device
- Basic and assembly language drivers supplied
- Meets IEEE 696 specification
- Industrial quality burned in and tested up to 125K bytes/sec under software control 3 parallel ports (8255-5)
- \$375



D&W DIGITAL, INC.
20655 Hathaway Avenue
Hayward, California 94541
(415) 887-5711

```

ltblk = 0;
maxblk=0;
return x;
}
x = getary(); /* load alternate tables */
if (x)
    return x;
altnflg = 0;
x = erropt(); /*set Adaptec option for error on successfull ECC */
return x;
}

```

Listing 3

```

/* dtcio rev 2.11.03 850612 Hul Tytus */

#define BASE 0x70      /* base i/o port */
#define BUSY 0x8        /* busy line */
#define GROUP 0x70      /* lines that cannot change in transfer */
#define STAT 0x12       /* pattern for status */
#define CMD 0x50        /* pattern for commands */
#define RD 0           /* pattern for read data */
#define WT 0x40        /* pattern for write data */
#define ERSTAT 2        /* status report line in data buss */
#define DMAEABL 3       /* enables dma and data */
#define DATEBL 2        /* enables data */
#define DONE 1          /* DMA operation over */
#define REQ 0x80        /* request line */
#define STATP BASE + 2  /* status port */
#define DATAP BASE      /* data port */
#define SELP BASE + 1   /* select port */
#define DMAD BASE + 3   /* dma address port */
#define DCCS BASE + 1   /* completion status port */
#define TARG 0x1         /* target device number */
#define SEL 0x42        /* select line */
#define SELOFF 0x2       /* software error */
#define SOFTER 0xf       /* drive not ready error */
#define DNR 4            /* drive not ready */
#define debug 0
#define detest 0

int getcon()           /* get controllers attention */
{
    unsigned x, time;
    x = 1;
    time = 65000L;
    while (x)           /* wait till not busy */
    {
        x = in(STATP);
        x = x & BUSY;
    }
    out(DATAP, TARG);  /* set data bit for target */
    out(SELP, SEL);    /* set select line */
    while (x == 0 && (time)) /* now wait till busy */
    {
        --time;
        x = in(STATP);
        x = x & BUSY;
    }
    if (!time)           /* if time expired, return drive not ready */
    {
        out(DATAP, 0);  /* clear bus lines */
        out(SELP, 0);
        return (DNR);
    }
    out(SELP, DATEBL); /* enable data and clear select line */
    return (0);
}

int putcmd(cmd)        /* send command to controller */
char *cmd;
{
    register int x;
start:
    x = request();
    x = x & CMD; /* make sure target is expecting a command byte */
    if (x != CMD)
        return (x);
    out(DATAP, *cmd); /* send byte of command string */
    ++cmd;
    goto start;        /* once again */
}

int status()           /* get status */
{
    register int x, y;
    x = in(DATAP);   /* read status byte */
    request();
    y = in(DATAP);   /* read and discard message byte */
    return (x);
}

int putdat(str)        /* send data to controller from str */
char *str;
{
    register int x;

```

Periscope Is On Sale!

```

start:
x = request();
x = x & CMD; /* make sure target is still expecting data */
if (x != WT)
    return (x);
out(DATAP, *str); /* send data */
++str;
goto start; /* once again */

int getdat(str) /* move data from controller to str */
char *str;
{
register int x;
start:
x = request();
x = x & CMD; /* make sure target is still expecting data */
if (x != WT)
    return (x);
out(DATAP, *str); /* send data */
++str;
goto start; /* once again */

int getdat(str) /* move data from controller to str */
char *str;
{
register int x;
start:
x = request();
x = x & CMD;
if (x != RD) /* be sure target has data to send */
    return (x);
*str = in(DATAP); /* read data */
++str;
goto start; /* once again */

int request() /* wait for request */
{
register int x, z;
x = 0;
while (!x) /* wait till request line goes high */
{
    x = in(STATP);
    z = x;
    x = x & REQ;
}
return (z);
}

int sense() /* get error code */
{
register int x;
static char cmd[6] = {3, 0, 0, 0, 0, 0};
static char res[6];
x = getcon();
if (x)
    return (x);
putcmd(&cmd[0]);
x = getdat(&res[0]);
status();
x = res[0] & 0x7F;
return (x);
}

int placef(num, dest) /* places long at dest in msb first order */
char *dest;
long num;
{
register int x;
dest = dest + 3;
x = 4;
while (x)
{
    *dest = num & 0xFF;
    num = num >> 8;
    --dest;
    --x;
}
}

long readc() /* get capacity of drive */
{
static char cmd[10] = {0x25, 0, 0, 0, 0, 0, 0, 0, 0, 0};
static char res[8];
register unsigned x;
long size;
x = getcon();
if (x)
{
    size = x;
    return (size);
}
putcmd(&cmd[0]);
getdat(&res[0]);
x = status();
if (x)

```



"A marvelous debugging tool..." says Peter Norton.

SUMMER SALE. Through August 31, the new software-only Periscope II-X is yours for just \$95! Periscope II, normally \$145, is only \$125!

NEW MODEL. Periscope I includes both a break-out switch for "spontaneous" debugging and a RAM board to protect the debugger from programs that overwrite memory. Periscope II includes a break-out switch that doesn't require an extra slot to install. And the new Periscope II-X is Periscope II without the break-out switch. So you can now get Periscope with no hardware at all!

NEW RELEASE. The new version 2.1 enhances all models with many of the features users have requested. Call for details.

CHECK OUT THE REVIEWS.

"Periscope strikes a balance of power and features that will please serious PC programmers."

—Jeff Duntemann, PRODUCT OF THE MONTH, PC Tech Journal, 1/86

"Periscope's diverse features, affordable price, and portability place it in a class by itself." —Ward Christensen, "Breaking out with Periscope," PC Tech Journal, 3/86

"Periscope represents the finest software debugger available in its class." —Andrew Fried, Computer Shopper, 4/86

REQUIREMENTS. An IBM PC, XT, AT or close compatible; 128K RAM; DOS 2.0 or later; one disk drive; and an 80-column monitor.

Periscope II-X, Software Only \$ 95*
Periscope II, with Switch \$125*
Periscope I, with Board & Switch .. \$295

*Prices good through August 31, 1986

Take advantage of the low sale prices while they last . . . Order your Periscope today! Call toll-free:



800/722-7006



30-Day Money-Back Guarantee

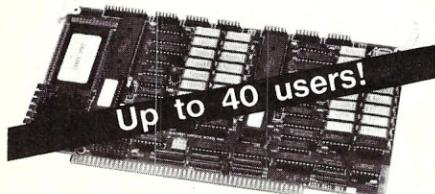
Shipping-\$3 UPS ground; \$5 COD; \$8 Canada; \$24 Foreign. We accept qualified PO's in the U.S.

Get Your Programs Up and Running;

UP PERISCOPE!

Data Base Decisions • (404) 256-3860
 14 Bonnie Lane • Atlanta, GA 30328

Products With Expandability

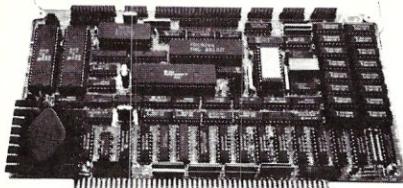


A two user Slave card based on Hitachi's Z80 compatible high speed, 10MHz super microprocessor.

Priced at **\$495^{00*}**

Features Include...

- 4-10 MHz Z80 Compatible HD64180
- 1/2 Megabyte Nonbanked Memory
- 2 Asynchronous Serial Ports To 38.4
- 1 High Speed Synchronous Port
- All Transfers Via 1.6 MHz DMA!!!
- Unique Expansion Port Offers;
 - 2 Additional Serial Ports or ...
 - 2 Parallel Ports or ...
- Real Time Clock With Battery Backup



The industry's fastest 8-bit Master CPU card with features superior to most 16-bit cards.

Priced at **\$495^{00*}**

Each Master Features . . .

- 4-10 MHz Z80 Compatible HD64180
- 1/2 Megabyte Nonbanked Memory
- 2 Asynchronous Serial Ports To 38.4
- 1 High Speed Synchronous Serial Port
- 4 Bi-directional Parallel Ports
- TurboDOS**, ZSYSTEMS**, CP/M** & OASIS** Operating Systems
- FDC Simultaneously Controls 8", 5 1/4", & 3 1/2" Drives
- SASI/SCSI Interface
- Optional High Speed Hard Disk/File Access Tape Backup and True ETHERNET Controller

*Prices apply to 6 MHz, 64KB versions and are good for a limited time only on purchases of ten or more. For less than ten, please call.

**Trademarks: TurboDOS - Software 2000; ZSYSTEMS - Echelon; CP/M - Digital Research; OASIS - THEOS Software

ID INTELLIGENT COMPUTER DESIGNS CORP.
23151 Verdugo Drive, Suite 113
Laguna Hills, CA 92653
(714) 581-7500

```

    {
        x = sense();
        return (0);
    }
    size = 0;
    while (x < 4) /* convert SCSI MSB first format to Intel format */
    {
        size = size << 8;
        size = size + res[x];
        ++x;
    }
    return (size);
}

int vfv(sect, cnt) /* verify cnt sector(s) - used to wiggle head */
long sect;
unsigned cnt;
{
    register int x;
    static char cmd[10] = {0x2f, 0, 0, 0, 0, 0, 0, 0, 0, 0};
    cmd[8] = cnt & 0xFF;
    cmd[7] = cnt >> 8;
    placef(sect, &cmd[2]);
    x = getcon();
    if (x)
        return (x);
    putcmd(&cmd[0]);
    x = status();
    if (x)
        x = sense();
    return (x);
}

/* move head to track specified in mode select */
/*
park()
{
    register int x;
    static char cmd[6] = { 0xb0, 0, 0, 0, 0, 0 };
    x = getcon(); if(x) return(x);
    putcmd(&cmd[0]);
    x = status();
    if (x) x=sense();
    return(x);
}
*/
int erropt()
/* set error handling options to indicate when the ECC is used */
{
    static char cmd[6] = {0xd0, 0, 0, 0, 4, 0};
    static char grp[4] = {0x65, 0, 1, 0};
    register int x;
    x = getcon();
    if (x)
        return (x);
    putcmd(&cmd[0]);
    x = putdat(&grp[0]);
    x = status();
    if (x)
        x = sense();
    return (x);
}

int reinit() /* tell controller to reinitialize */
{
    static char cmd[6] = {0xd0, 0, 0, 0, 4, 0};
    static char grp[4] = {0x60, 0x30, 0, 0};
    register int x;
start:
    x = getcon();
    if (x)
        return (x);
    putcmd(&cmd[0]);
    x = putdat(&grp[0]);
    x = status();
    if ((x & 8)) /* if check byte busy bit is set, try again */
        goto start;
    if (x)
        x = sense();
    return (x);
}

int test() /* see if drive is ready */
{
    register int x;
    static char cmd[6] = {0, 0, 0, 0, 0, 0};
    x = getcon();
    if (x)
        return (x);
    putcmd(&cmd[0]);
    x = status();
    if (x)
        x = sense();
    return (x);
}

```

```

int dread(sect, point, seg, cnt) /* read cnt sectors via DMA to seg:point */
long sect;
unsigned point, seg;
unsigned cnt;
{
    register int x;
    static char cmd[10] = {0x28, 0, 0, 0, 0, 0, 0, 0, 1, 0};
    placef(sect, &cmd[2]);
    cmd[8] = cnt & 0xFF;
    cmd[7] = cnt >> 8;
    x = dmaio(point, seg, &cmd[0]);
    return (x);
}

int dwrite(sect, point, seg, cnt) /* write cnt sectors via DMA from seg:point */
long sect;
unsigned point, seg;
unsigned cnt;
{
    register int x;
    static char cmd[10] = {0x2e, 0, 0, 0, 0, 0, 0, 0, 1, 0};
    placef(sect, &cmd[2]);
    cmd[8] = cnt & 0xFF;
    cmd[7] = cnt >> 8;
    x = dmaio(point, seg, &cmd[0]);
    return (x);
}

int dmaio(bpoint, seg, cmd) /* read or write data from/to seg:bpoint via DMA */
unsigned bpoint, seg;
char *cmd;
{
    register int x, z;
    char c;
    long add;
    x = getcon();
    if (x)
        return (x);
    out(SELP, DMAEBL); /* enable DMA */
    add = seg; /* convert 2 register format to 24 bit pointer */
    add = add << 4;
    add = add + bpoint;
    c = add >> 16;
    out(DMAD, c);
    c = add >> 8;
    out(DMAD, c);
    c = add & 0xFF;
    out(DMAD, c);
    z = 10;
    while (z--) /* send command - count the bytes this time to cover */
        {
            /* for a DTC quirk */
            x = request();
            x = x & CMD;
            if (x != CMD)
                break;
            out(DATAP, *cmd);
            ++cmd;
        }
    x = 0;
    while (!x) /* wait till DMA is done */
        {
            x = in(STATP);
            x = x & DONE;
        }
    /* get status byte from host adapter - another DTC quirk */
    x = in(DICCS);
    if (x)
        x = sense();
    return (x);
}

```

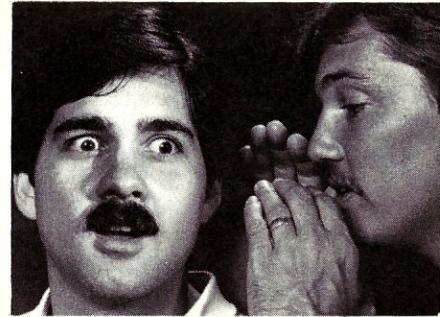
Listing 4

*** MSDOS code that points to Maximum Sector Size Word ***

```

4000:406D 53      PUSH   BX
4000:406E 51      PUSH   CX
4000:406F 52      PUSH   DX
4000:4070 E87CCC  CALL   OCEF
4000:4073 26      ES:    ; load ax with boot drive's
4000:4074 8B4602  MOV    AX,[BP+02] ; sector size
4000:4077 36      SS:    ;
4000:4078 3B060101 CMP    AX,[0101] ; compare with MSSW
4000:407C 7604      JBE   4082   ; if greater,
4000:407E 36      SS:    ;
4000:407F A30101  MOV    [0101],AX ; increase MSSW
4000:4082 5A      POP    DX
4000:4083 59      POP    CX
4000:4084 5B      POP    BX
4000:4085 8CD8  MOV    AX,DS
4000:4087 5E      POP    SI
4000:4088 1F      POP    DS
4000:4089 26      ES:    ;
4000:408A 897612  MOV    [BP+12],SI

```



If You Could Hear The Language They're Using!

What they're saying would astound you.

They've switched from BASIC, and PASCAL, and a host of other languages and even stand-alone database programs.

They've switched to COMP Computing Standard MUMPS...CCSM, the Database Language.

"The best software product of its kind, that I have come in contact with." *Computer Language Magazine*

"I've found the best in CCSM...fast in development and fast in execution...no data-typing problems, no concerns for program size, no concerns for file or device opens..." R.D. Ashworth, Ph.D.

"...5 years in Basic, Pascal, C, dBase, and Dataflex... I have never worked with a language/programming environment as responsive, easy to use and as powerful as CCSM" P.K. Wayne, MD, Ph.D.

CCSM, the Database Language is a powerful, flexible string-oriented language, with the features of a complete development and operating system. Typical programs are written in 1/3 to 1/5 the amount of code, compared to BASIC or PASCAL.

Multi-Tasking Available, Too!

CCSM, the Database Language offers a true multi-tasking option. Run multiple, concurrent background processes for data searches, report generation, etc.

Order CCSM, the Database Language today.

In a very short time, you'll be surprised at the language you're using.

IBM's and compatibles...128K min.

1-800-257-8052

in Texas 713-529-2576

MGlobal

1601 Westheimer, Suite 201

Houston, TX 77006

AMEX, VISA AND MC accepted by phone

Single User disk and operations manual \$ 59.95
"Cookbook of MUMPS" (includes disk) \$ 24.95

SPECIAL Single user & "Cookbook" \$ 75.90

Multi-tasking \$ 149.95

Disks are *Programmer's Toolkit* \$ 49.95
non-copy- *Graphics option* \$ 49.95
protected *Multi-user* \$ 450.00

MacMUMPS (Macintosh version) \$ 89.95
shipping and handling \$ 3.00

Texas residents add 6 1/8% sales tax

*IBM is a trademark of International Business Machines;
Macintosh is a trademark licensed to Apple Computer"*

MORE... VALUE and PERFORMANCE

with Mitek's

Relocatable Z80 Macro Assembler and Z80 Symbolic Debugger

NEW Z80 Symbolic Debugger

- Only \$49.95 plus shipping.
- Screen oriented with a simultaneous display of instruction mnemonics, register, stack, and memory values.
- Breakpoints may be set on any combination of fixed memory address, register values and/or memory values.
- Uses Digital Research compatible SYM files.
- Supports Hitachi HD64180.

Relocatable Z80 Macro Assembler

- Only \$49.95 plus shipping.
- 8080 to Z80 Source Code Converter.
- Generates Microsoft compatible REL files or INTEL compatible hex files.
- Compatible with Digital Research macro assemblers MAC & RMAC.
- Generates Digital Research compatible SYM files.
- Conditional assembly.
- Phase/dephase.
- Cross-reference generation.
- Full Zilog mnemonics.
- INCLUDE and MACLIB FILES.
- Separate data, program, common, and absolute program spaces.
- Supports Hitachi HD64180.
- Z80 Linker and Library Manager for Microsoft compatible REL files available as an add-on to Assembler.

ATTENTION Turbo Pascal Users:

Assembler will generate Turbo Pascal in-line machine code include files.

TO ORDER, CALL TOLL FREE: 1-800-367-5134, ext. 804

For information or technical assistance: (808) 623-6361

Specify desired 5 1/4" or 8" format. Personal check, cashier's check, money order, VISA, MC, or COD welcomed.

P. O. Box 2151
Honolulu, HI 96805

MITEK

Z80 is a trademark of Zilog, Inc. MAC and RMAC are trademarks of Digital Research, Inc. Turbo Pascal is a trademark of Borland International, Inc.

PRICE LIST

Z80 Macro Assembler:	\$49.95
Assembler, Linker, and Library Manager:	\$95.00
Manual Only:	\$15.00
Z80 Symbolic Debugger:	\$49.95
Manual Only:	\$15.00
Assembler, Linker, Library Manager, and Debugger:	\$134.95

Include \$5 for shipping and handling.

What you see is what you get . . . and send!

Transfer Protocol: Modem7/CRC Packet Size: 128 Files: 1

Block #	of	Kbytes	%	Time Remaining	Consec	Error File	Errors Total	Total Kbytes
31	522	3	5	5:06	8	0	0	3

Errors:

Status: Transfer in progress



[CTL-C to abort]

Sending: >ANYFILE.AQC

Announcing Version 1.6 of **MEX**, the communications software with a view from the top. Regardless of your level of sophistication, **MEX** can put you on top of the data transfer game and keep you there. For the executive on the go, our new pull-down transfer screen and easy-to-use menus reduce the complexities of modem communications to a few keystrokes. For the advanced user, **MEX**'s greatly enhanced script processor offers a complete programming language for development of highly secure custom applications. If communication is money in your business, **MEX** may be the best investment you make this year.

Two options available:

MEX-PC is the most complete modem software you can buy. Allows you to switch between menu-driven and command-driven communications at will. Makes full use of Hayes AT command set, with overlays available for most other modems. Features include: complete script processor programming language; user-definable keystrings; auto-dial and auto-baud-set phone libraries; all popular protocols, including MODEM-7 batch transfers. **\$59.95***

XMODEM CRC/XMODEM CHECKSUM/KERMIT/COMPUSERVE A/128 or 1K BLOCKS

MEX-PAC — All the features of **MEX-PC**, plus:

- (A) remote module that allows you to run your office computer from home, and vice versa; and
- (B) terminal emulation that lets your PC masquerade as a DEC VT52/100 or Televideo 925 terminal for on-line communication with mainframes. **\$99.95***

Versions available for IBM-PC and compatibles, Tandy 2000 and most CP/M machines.

MasterCard, VISA welcome

**Give us a toll-free call at
1-800-NITEOWL**

In Wisconsin, 1-414-563-4013

NightOwl Software, Rt. 1 Box 7, Ft. Atkinson, WI 53538

* plus shipping. Wisconsin residents add 5% sales tax.

YOUR EAST COAST HI-TECH CENTER

QUALITY
COMPUTER
PRODUCTS



DIGITAL
DECISIONS
SYSTEMS

FOR THE
BUSINESS
PROFESSIONAL

Compu Pro/Viasyn-S100

CPU 286-8 3CY	542	SYS.SPT.1	319	RAM 23 128K	262
CPU 286-8 2CY	674	INTF 4	333	RAM 24 1MEG	2249
CPU 86 -10	295	DISK 1A	445	MDRIVE/H.5M	366
CPUZ-8	209	RAM 22 256K	446	MDRIVE/H 2M	819
SPUZ-256	289	RAM 23 64K	187	INTF 3-8	445
				DISK 3	520

LOMAS MEGARAM 2-4 MEGABYTES

MODIFIED FOR USE WITH DISK 3 SYSTEMS

ADM 12+ 2 PAGES	489	MAXTOR	1085	1889	TANDON TM755
LINK 125 2 PAGES	509		1140	2669	51 MEG - 20 MS 945

We can custom configure any system to meet your needs. Installation & training available in some areas.

WATCH FOR OUR 60 MEG S100 TAPE BACKUP...

CALL!!

DD8000 series Professional Computer
8 MHz V-20 CPU Provides Superior Performance
At A Reasonable Cost!

STANDARD FEATURES INCLUDE:

- 1 MEGABYTE RAM
- 6 EXPANSION SLOTS
- 2 PRINTER PORTS
- IBM's PC-DOS 3.1
- 2 360K FLOPPY'S
- ADVANCED COLOR/MONO/HERCULES GRAPHICS DISPLAY ADAPTER
- 2 SERIAL PORTS
- MONO TTL MONITOR
- OPTIONAL HARD DISKS – 20 TO 56 MEG
- SASI PORT

22 MEG \$2639

33 MEG \$2889

56 MEG \$3779

\$1899

FLOPPY ONLY

WILL SUPPORT UP TO THREE PC-DOS USERS WITH CCP/M-XM

P.C. UPGRADES

MF-384 384K RAM & SER/PAR/CLK/GAME	229
MULTI 1/0 SER/PAK/CLOCK HALF CARD	169
QUAD 1/0 4 RS232 SER FOR PC OR AT	199
SYNCO TTL MONITOR - TILT/SWIVEL - A OR G	135
TOSHIBA 360K FLOPPY W/MTG HDWE	95
HARD DISKS	
TULIN 26 MEG PC 389 40 MEG PC 499	
40 MEG AT SEAGATE 10 MEG PC CALL FOR PRICE	
CONTROLLER FOR ABOVE — WICABLES 179	
ADAPTEC 2070 2-7 RLL HARD DISK CONTROLLER 289	
INCREASE SPEED AND CAPACITY BY 50%	
LIMITED IBM PC W/ 512K RAM • MONO TTL \$1995	
2 • 360K FLOPPYS SER • PAR • CLOCK • GAME • VIDEO • DOS RGB COLOR \$2250	
OFFER!	

Multitech
POPULAR 500
COLOR PC SYSTEM
\$1299

- 2 360 K FLOPPYS
- 512K RAM
- PARALLEL PORT
- DOS 2.11
- RGB MONITOR
- RS232 SERIAL
- GAME PORT
- WORD PROCESSOR

QUALITY AND ECONOMY IN 16 BIT COMPUTING

OUTPUT TECHNOLOGY CORPORATION 700 CPS!

High Speed Matrix Printers

MODEL OT-700 \$1695

OT 700e \$1895

BULK 5.25" DISKETTES By BROWN
 CERT. DS/DD W/ HUB RINGS & TYVEC SLEEVES
 LIFETIME WARRANTY PKG 20 \$.79 ea. BOX .65 ea.
 CALL FOR VOLUME PRICING

GIFFORD MC-DOS USERS MC-DOS ENHANCEMENTS!

UPGRADE YOUR MC-DOS SYSTEM
 TO SUPPORT HIGH CAPACITY HARD DISKS

**75 TO 330 MEGABYTES
 PATCH KIT 1 — SOFTWARE ONLY \$100**

(YOU SUPPLY HARD DRIVE)

COMPLETE KITS AVAILABLE — \$ CALL \$

STANDARD KITS ARE CONFIGURED FOR

MAXTOR 1000 SERIES DRIVES.

CUSTOM CONFIGURATIONS ARE AVAILABLE.

REQUIRES DISK 3 CONTROLLER SYSTEM.



DIGITAL DECISIONS SYSTEMS

229 LAKEVIEW AVE • CLIFTON, NJ 07011

1-800-862-6233

Wait for tone,
 dial Ext. 0014762



ALL PRICES SUBJECT TO CHANGE — SHIPPING EXTRA

IBM is a Reg. TM of International Business Machines, CP/M, Concurrent DOS are Reg. TM of Digital Research Inc. MSDOS is a TM of Microsoft. CompuPro is a Reg. TM of Viasyn Corp. OT 700 is a Reg. TM of Output Technology Corp. Popular 500 is a Reg. TM of Multitech Int.

Program Interfacing To MS-DOS

by William G. Wong

This article presents a character device driver which replaces the DOS printer device driver. The driver assumes ROM BIOS support of interrupt 17 (hex) for printer output. The driver replaces the standard DOS printer device named PRN. It can be accessed by any program which normally uses the PRN device for printer output including the DOS COPY and PRINT commands.

Unfortunately, this replacement will not be used by all programs because some go directly to the ROM BIOS by passing both DOS and the device driver. You may be surprised which programs do perform such nefarious tasks. It is possible to interpose a driver-like program between the ROM BIOS driver and any program which would access the driver directly but this is not within the scope of this article.

The printer driver is very simple since it is an output only device which essentially translates DOS requests to ROM BIOS calls. Even so, a very simple twist has been added which may prove useful for some and can be extended in a variety of ways. The example also translates easily into other character oriented device drivers.

Normally, a printer driver accepts characters from DOS and sends them to the printer interface. This driver also performs a very simple translation. Any tilde characters, ~ or 7E hex, are converted to the escape (ESC) character, 1B hex. The purpose of such a translation is to allow text files to be generated by a text editor which may not be able to print the ESC character. Text files may also be displayed using the DOS TYPE command. Embedding ESC characters within a text file which is displayed on the main console can have interesting effects, especially if ANSI.SYS is loaded.

A sample text file for an Epson LQ-1500 printer is:

- ~MELITE text
- ~PPICA text
- ~4Italic text
- ~5Normal text

where each line is preceded by a two character escape sequence, such as ~M, which is printed as an ESC character followed by the letter M. This is not actually printed by the printer but rather tells it to print subsequent text using the ELITE font. The control sequences may appear at any point within the text.

Part IX A Printer Device Driver

Although some text editors allow embedded control characters, these tend to display each character as two characters, normally an up arrow (^) followed by the corresponding control character. The ESC character comes out as ^[. However, some text editors also convert the character to the two character display sequence when the file is printed thereby defeating the embedded character usage.

The tilde character was chosen because it is not often used, although some language implementations use it as an operator. This would probably obviate the use of this device driver, although another could easily be implemented with other characteristics.

In fact, this approach can be used for some interesting applications. For example, instead of just translating the tilde character to an ESC character, how about translating the control sequences for a standard printer, like the FX-80, into corresponding control sequences for another printer. This allows a new printer to be supported by an application even though the application does not support such a device. Although this requires the creation of a device driver, it may be a viable alternative in many situations. It is also a more general solution to the question of customizing applications.

The implementation of such a device driver is relatively straight forward but may take a great deal of program code and table space. Such a driver was not presented because the translation mechanism would probably be more complex than the printer driver presented here.

In general, such a driver can be implemented as a co-routine or by a table driven state machine. A co-routine requires a separate stack which is used by the translation process. A stack switch is performed with each character to be printed. The process then continues based

upon the sequence of characters received. This approach has the advantage of being easy to understand and extend, once the stack switch mechanism is in place. A table driven state machine takes each character and indexes a table of characters. The entry which has a matching character indicates the next state and a possible set of operations to be performed. For example, the corresponding control sequence could be sent to the real printer after the proper untranslated control sequence is received from DOS.

In either case, certain control sequences are translated while other characters, usually displayable text, are sent to the printer without translation. The possibilities are endless.

This approach is not restricted to printer drivers. Any character device can be developed in a similar fashion including input devices. Translation of function keys using a resident programming facility are also possible as well as communication drivers which can guarantee error free transmission between devices. In this case, characters are normally buffered into a packet which is sent to the other computer along with error detection information. The packet is re-sent until the other computer verifies proper reception. This is similar to file transfer protocols used by communication programs.

THE PRINTER DEVICE DRIVER

But back to the device driver at hand. The source code for this sample driver is included at the end of the article and is very similar to the one described in the prior article which presented a memory disk block device driver. The source code is divided into four sections. The first contains constant definitions including offsets to various structures and values which may be placed into these structures. This information is device independent. The second section defines common structures and variables used by all device drivers. The Bios Parameter Block (BPB) structure used in the prior article is removed since the printer driver is a character device. The third section contains source code which is common to all device drivers. The first three sections have been extracted from the memory disk device driver with the exception of the changes to the device header which is changed to reflect a character de-

vice (a_character) instead of a block device (a_block). Also, the name field is now used for the name of the printer (PRN) instead of containing the number of block devices. The fourth section contains device specific code which is described in the following section.

PRINTER DEVICE SPECIFIC CODE

The device specific functions supported by the printer driver are:

Function	Description
initialize	setup printer
write	write characters
write_verify	write and verify operation
output_status	check output status
output_flush	flush output (null operation)

The functions not supported by this driver are:

Function	Reason
build_bpb	block device only
media_check	block device only
ioctl_read	no purpose
ioctl_write	no purpose
read	input character device only
check_input	input character device only
input_status	input character device only
input_flush	input character device only

The 'initialize' function is much shorter than the corresponding memory disk function since there is no device or internal parameters to initialize. There is no computation of the size of the device driver resident code since the printer driver is fixed unlike the memory disk driver.

The 'write' and 'write_verify' functions are identical since there is no verification using a printer. Also, the latter is not accessible in normal operations since the verify mode is normally reserved for block devices. Direct access to the device driver is necessary for a program to access this function.

The 'output_status' function simply converts the results from the ROM BIOS status call into the corresponding status word required by DOS. This is normally used in an application which polls a device so it will not lock up if the device is not ready.

The 'output_flush' function performs no operation because the ROM BIOS printer interface has no corresponding operation. This function normally clears any buffer or stops any control operation. For example, an interrupt driven print spooling device driver would have an internal buffer which might be relatively large.

The 'read', 'check_input', 'input_status' and 'input_flush' functions were not implemented because the printer does not act as an input device. A communications device would need these functions to be implemented within the device specific section.

MAKING CHANGES

Adding single character translations similar to the tilde/ESC character conver-

```

; === Loadable Printer Device Driver ===
;      v1.00 05-15-86 Initial version
; by: William G. Wong
; Logic Fusion, Inc.
; 1333 Moon Drive
; Yardley, PA 19067
; =====

; == Constant Definitions ==
;

; --- Request Header Structure Definitions ---
rh_length equ 0 ;(byte) length field
rh_unit equ 1 + rh_length ;(byte) unit field
rh_command equ 1 + rh_unit ;(byte) command code
rh_status equ 1 + rh_command ;(word) status field
rh_reserved equ 2 + rh_status ;(8 bytes)
rh_size equ 8 + rh_reserved ;request header size

; --- Request header status values ---
s_done equ 0100h ;done, no errors
s_busy equ 0200h ;busy, no errors
e_protect equ 8000h ;error: write protect
e_unknown_unit equ 8001h ;error: unknown unit
e_not_ready equ 8002h ;error: not ready
e_command equ 8003h ;error: unknown command
e_crc equ 8004h ;error: bad CRC
e_bad_length equ 8005h ;error: bad structure length
e_seek equ 8006h ;error: bad seek
e_media equ 8007h ;error: unknown media
e_not_found equ 8008h ;error: sector not found
e_paper equ 8009h ;error: out of paper
e_write equ 800ah ;error: write fault
e_read equ 800bh ;error: read fault
e_general equ 800ch ;error: general error not listed above

; --- Non-destructive read parameter block ---
rh_read_data equ rh_size ;(byte) non-destructive data

; --- Input/output parameter block ---
rh_media equ 0 + rh_size ;(byte) media descriptor
rh_buf_offset equ 1 + rh_media ;(word) transfer buffer offset
rh_buf_segment equ 2 + rh_buf_offset ;(word) transfer buffer segment
rh_buf_size equ 2 + rh_buf_segment ;(word) transfer buffer size
rh_start equ 2 + rh_buf_size ;(word) transfer starting sector
rh_buffer equ rh_buf_offset ;(dword) buffer address
m_fixed equ -0F8h ;media: fixed disk
m_ss9 equ 0fh ;media: single sided, 8 sectors/track
m_ds9 equ 0fdh
m_ss8 equ 0feh
m_ds8 equ 0ffh

; --- Build BPP parameter block ---
;preceeded by media descriptor
rh_bpb equ 1 + rh_media ;(dword) bpb buffer address
rh_tbl_offset equ 4 + rh_bpb ;(dword) bpb table offset/segment
rh_tbl_segment equ 2 + rh_tbl_offset

; --- Media Check parameter block ---
;preceeded by media descriptor
rh_check equ 1 + rh_media ;(byte) media check result
mc_changed equ -1 ;media has changed
mc_maybe equ 0 ;media may have changed
mc_same equ 1 ;media has not changed

; --- Initialize parameter block ---
rh_units equ 0 + rh_size ;(byte) number of units supported
rh_end_offset equ 1 + rh_units ;(word) end address of driver
rh_end_segment equ 2 + rh_end_offset
rh_bpb_offset equ 2 + rh_end_segment ;(word) BPP array address
rh_bpb_segment equ 2 + rh_bpb_offset

; --- Device Driver Header Attribute Definitions ---
a_input equ 0001h ; standard input device
a_output equ 0002h ; standard output device
a_nul equ 0004h ; NUL device
a_clock equ 0008h ; CLOCK$ device
a_dos equ 0 ; DOS block device (bit 13)
a_not_dos equ 2000h ; non-DOS block device (bit 13)
a_ioctl equ 4000h ; IOCTL functions supported
a_block equ 0 ; block device (bit 15)
a_character equ 8000h ; character device (bit 15)

; === Device Driver Header Definition ===
cseg segment para public 'CODE'

driver proc far
assume cs:cseg,ds:cseg,es:cseg

dd    -1          ; last driver in chain
dw    a_character ; driver attribute
dw    dev_strategy ; offset to strategy routine
dw    dev_interrupt ; offset to interrupt routine
db    'PRN'        ; number of devices or device name

```

sion are relatively easy, as well as single character to multiple character sequences, such as converting a form feed to a control sequence for a daisy-wheel printer. Converting multiple character sequences into multiple character sequences requires the use of co-routines or state tables, as mentioned earlier.

Using the driver as a basis for a two way driver for a serial port is also straight forward, assuming the procedures for receiving and sending a character are already available. The input functions would have to be removed from the unimplemented definitions as the corresponding support code is added.

One interesting possibility is to make a memory disk-like character device which stores all output in a memory buffer and empties the buffer when information is read from the device. This is one possibility for applications which must use a specific device, since the results could be copied after the application is finished running.

SUMMARY

The character device driver is a simple entity to build assuming the support functions are already available. Like any device driver, the hardest part is making sure these support functions work properly before integrating them into the system, since it is almost impossible to debug new drivers interactively in the same sense that normal programs run using a debugger such as DOS DEBUG.

Having a device driver shell makes it much easier to develop a device driver. Separating the device specific code into a separate file and linking the driver together leads to the possibility of creating a separate front end to test new device specific functions. This front end would call the initialization function and then call the other functions, possibly using an interactive user interface.

This wraps up device drivers. The next article will discuss the changes and additions between DOS 2.x and DOS 3.x. Some of the additions relate to device drivers.



Bill Wong is the President of Logic Fusion, Inc., 1333 Moon Drive, Yardley, PA 19067, a systems software development firm.

```

; === Device Driver Tables ===
;
; --- Request Header Address set by dev strategy ---
rh_address dd 1 dup (?) ; request header base address
rh_offset equ word ptr rh_address
rh_segment equ word ptr ( rh_address + 2 )

; --- Request Header Command Dispatch Table ---
cmd_table dw initialize ;initialize driver
dw media_check ;media check
dw build_bpb ;build BPB
dw ioctl_read ;IOCTL read
dw read_normal ;normal read
dw check_input ;non-destructive read/status
dw input_status ;input status
dw input_flush ;flush input buffers
dw write_normal ;normal write
dw write_verify ;normal write with read verify
dw output_status ;output status
dw output_flush ;flush output buffers
dw ioctl_write ;IOCTL write

; === Common Device Driver Routines ===
;
; --- Device Driver Strategy Routine ---
;
; es:bx == request header address
dev_strategy proc far
    mov cs:rh_offset,bx
    mov cs:rh_segment,es
    ret
dev_strategy endp

; --- Device Driver Interrupt Routine ---
dev_interrupt proc far
    push ax ; save registers used
    push bx
    push cx
    push dx
    push di
    push si
    push ds
    push es
    cld ;clear direction flag
    push cs ;setup small memory model
    pop ds ;ds := program segment
    les bx,rh_address ;es:bx := request header index
    mov si,es:rh_command[bx];si := request command (byte)
    and si,0ffh ;si := request command
    add si,si ;si := word table offset
    call word ptr cmd_table[si];ax := command result
    lds bx,cs:rh_address ;ds:bx := request header index
    mov rh_status[bx],ax ;update request status
    pop es ;restore registers
    pop ds
    pop si
    pop di
    pop dx
    pop cx
    pop bx
    pop ax
    ret
dev_interrupt endp

; **** END OF DEVICE INDEPENDENT PORTION OF DRIVER ****

; --- Printer Device Driver Definitions ---
printer equ 0 ; printer number (0-2)
printer_interrupt equ 17h ; BIOS printer interrupt

; --- Printer Device Driver Code ---
;
; --- Driver support functions (near functions) ---
;
; es:bx == request header
; ds == cs
; All other registers are usable.
;
; ax := result status
;
; --- Initialize driver ---
initialize proc near ; initialize driver
    lds bx,cs:rh_address; ds:bx := request header address
    mov word ptr rh_end_offset[bx],offset last
    mov rh_end_segment[bx],cs ;return ending driver address
    push cs ;use code segment as data segment
    pop ds
    mov dx,offset initok
    mov ah,9
    int 21h
    mov ax,s_done ; ax := done, no errors
    ret
initok db 'PRN device ~ to ESC translation installed',10,13,'$'
initialize endp

; --- Write to device ---
write proc near ; normal write
    mov cx,es:rh_buf_size[bx]; cx := number of characters
    lds si,es:rh_buffer[bx] ; ds:si := buffer address

```

```

write_loop:
    push si      ; save parameters
    push cx
    lodsb
    call write_char ; al := next character
    pop cx      ; ah := write result
    pop si      ; restore parameters
    test ah,29h ; zero := no timeout, no error, paper
    jnz translate_error ; skip if error occurred
    inc si      ; index next character
    loop write_loop ; loop if all characters transferred
    mov ax,s_done ; ax := transfer done
    ret
write endp

; ---- Write character to device after translation ----
; al = character (translate '' to ESC)
; ah := result
write_char proc near
    cmp al,'`' ; zero := translate '' to ESC
    jnz output_char ; skip if no translation
    mov al,lbh' ; al := ESC character
output_char:
    mov ah,0 ; ah := print character
    mov dx,printer ; dx := printer number
    int printer_interrupt ; ah := result
    ret
write_char endp

; ---- Return output status ----
output_status proc near ; output status
    mov ah,2 ; ah := get printer status
    mov dx,printer ; dx := printer number
    int printer_interrupt ; ah := printer status
    jmp translate_error ; ax := result status
output_status endp

; ---- Translate error value ----
; ah = printer status
; ax := device result word
translate_error proc near
    test ah,0lh ; zero := no timeout
    jnz err_timeout ; skip if timeout
    test ah,08h ; zero := not general error
    jnz err_write ; skip if general error
    test ah,20h ; zero := not out of paper
    jnz err_paper ; skip if out of paper
    mov ax,s_done ; ax := no errors
    ret
err_timeout: ax,e_not_ready
    ret
err_write: ax,e_write
    ret
err_paper: ax,e_paper
    ret
translate_error ndp

; ---- Flush output buffers ----
output_flush proc near ; flush output buffers
    mov ax,s_done ; ax := operation done
    ret
output_flush endp

; === Support Functions ===
; === Unimplemented functions ===
unimplemented proc near
    mov ax,e_command
    ret
unimplemented endp

write verify equ write ; write with verify
media_check equ unimplemented ; media check
build_bpb equ unimplemented ; build BPB
ioctl_read equ unimplemented ; IOCTL read
check_input equ unimplemented ; non-destructive read/status
input_status equ unimplemented ; input status
input_flush equ unimplemented ; flush input buffers
ioctl_write equ unimplemented ; IOCTL write
read equ unimplemented ; read

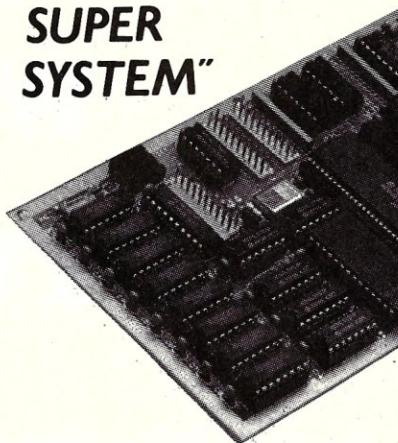
; === End of Driver ===
last equ $

; === End of Memory Disk Device Driver ===
driver endp
cseg ends
end

```

Byte Magazine called it.

"CIARCA'S SUPER SYSTEM"



The SB180 Computer/Controller

Featured on the cover of Byte, Sept. 1985, the SB180 lets CP/M users upgrade to a fast, 4" x 7½" single board system.

- 6MHz 64180 CPU
(Z80 instruction superset), 256K RAM, 8K Monitor ROM with device test, disk format, read/write.
- Mini/Micro Floppy Controller
(1-4 drives, Single/Double Density, 1-2 sided, 40/77/80 track 3½", 5¼" and 8" drives).
- Measures 4" x 7½", with mounting holes
- One Centronics Printer Port
- Two RS232C Serial Ports
(75-19,200 baud with console port auto-baud rate select).
- Power Supply Requirements
+5V +/- 5% @500 mA
+12V +/- 20% @40mA
- ZCPR3 (CP/M 2.2/3 compatible)
- Multiple disk formats supported
- Menu-based system customization

SB180-1
SB180 computer board w/256K bytes RAM and ROM monitor \$369.00

SB180-1-20
same as above w/ZCPR3, ZRDOS and BIOS source.....\$499.00

-Quantity discounts available-

NEW

COMM180-M-S
optional peripheral board adds 1200 bps modem and SCSI hard disk interface.

TO ORDER
CALL TOLL FREE **TELEX**
1-800-635-3355 **643331**

For technical assistance or
to request a data sheet, call:
1-203-871-6170



Micromint, Inc.
25 Terrace Drive
Vernon, CT 06066

Using SYMDEB with NMI Breakpoints

by James G. Owen

"NMI" stands for "non-maskable interrupt". The 8088 - like other microprocessors - has a pin with this name. When it sees a low-to-high transition on this pin, it will immediately stop what it's doing, save the flags and the current address on the stack, go directly to segment 0, offset 8, get the segment/offset address stored there, and start executing code at that address. It will do this no matter what it was doing when the NMI occurred, and regardless of whether interrupts are enabled or disabled.

SYMDEB ("Symbolic Debugger") is Microsoft's latest, new-and-improved version of the DEBUG program that comes with MS-DOS. I received it when I upgraded to the latest revision of the Microsoft assembler. SYMDEB has a number of enhancements, notably the symbolic reference capability (which works with Microsoft language products), but it also has the ability to use NMIs as an aid to debugging. If the application program has decided to stop doing *anything* - the machine is "hung up" - a debugger that uses NMI can break out of the hung-up state. Without such a feature, as most programmers know to their sorrow, the only way out is the reset button (or, on IBM computers, various mysterious key combinations and/or the AC power switch).

There are hardware parts available from IBM and elsewhere that provide access to the NMI in the IBM PC. To use SYMDEB's NMI facility, one is supposed to obtain one of these. I don't own an IBM PC (I couldn't get security clearance); there may be two or three others like me out there, and some PC users might like the NMI feature to work correctly. This article describes the minimal hardware, and provides a fix for SYMDEB's inept approach to the NMI handler.

Curing The NMI Problem In The MS-DOS Symbolic Debugger

SYMDEB NMI PROBLEMS

In a previous incarnation of Micro/Systems Journal I had a short piece about using NMIs with DDT in CP/M-80 computers. It appears, however, that the Microsoft programming staff did not pay careful attention. You can demonstrate SYMDEB's difficulties by appropriately invoking SYMDEB with NMIs enabled and some application program that, like most applications, calls the operating system for input from the user. Wait until the program is waiting for input. Actuating the NMI trigger at this time will probably get you into SYMDEB; using the "G" command to return to the application program will probably crash the system. On my computer (a Kaypro 4 with a Co-Power MS-DOS card) many amusing things are written all over the screen.

What went wrong is that the NMI occurred while the microprocessor was executing code in the operating system. As it says somewhere in the DEBUG manual, you can't trace or breakpoint inside MS-DOS - because MS-DOS isn't re-entrant, nor is DEBUG or SYMDEB - and doing so is likely to cause a crash. The NMI handler in SYMDEB should check for this; it doesn't.

The listing accompanying this article, MODNMI.ASM, is a fix for this; it checks the code segment of the program being NMIED, and does an IRET if it decides execution is inside the OS or SYMDEB itself. MODNMI is invoked from inside SYMDEB; it overlays SYMDEB's sign-on message and alters the NMI vector to point to the new code. After the new code checks the segment of the interrupted program, execution is transferred to SYMDEB's existing NMI handler if all is well; otherwise, control is returned to the interrupted program through an IRET.

USING MODNMI

MODNMI was created with SYMDEB version 3.00. Before it overlays the sign-on, it checks to see that the NMI vector is pointing where it expects it to point; if it isn't, it prints an error message and departs. With non-IBM computers, this could happen if the user forgot to use the NMI "/n" switch when invoking SYMDEB (SYMDEB always uses the NMI option in IBM computers). The error could also occur if the version of SYMDEB is different from the one I worked with.

Follow these steps to assemble and use MODNMI:

- 1) Enter source from listing and assemble with MASM: "A>MASM MODNMI;".
- 2) Link result (producing standard "no stack segment" warning): "A>LINK MODNMI;".
- 3) Convert .EXE file to a .COM file: "EXE2BIN MODNMI". Exe2bin will produce MODNMI.BIN.
- 4) Invoke SYMDEB with NMI option and load MODNMI.BIN: "A>SYMDEB /N MODNMI.BIN". When SYMDEB prompt appears, enter "G" command to execute MODNMI.BIN.

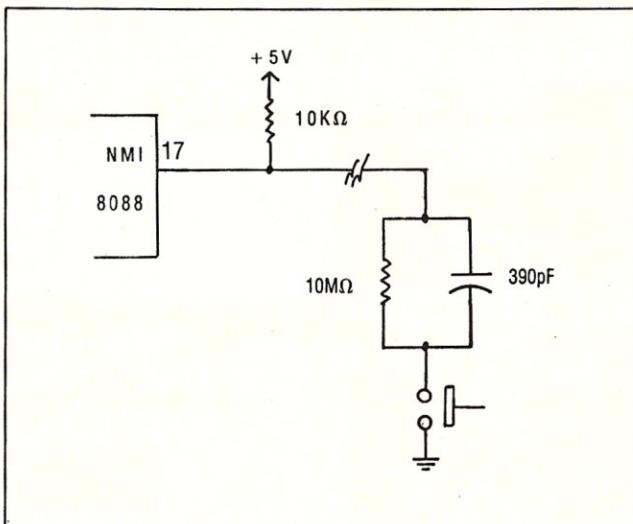


Figure 1. A very simple de-bouncing circuit to drive NMI. Values are approximate, and should be "tried-out" first.

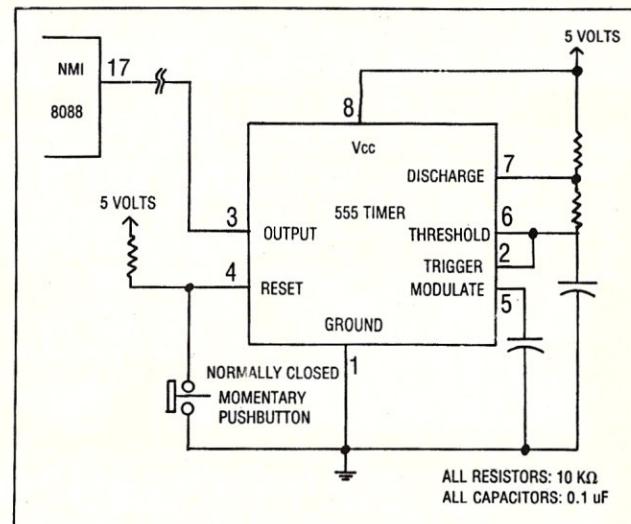


Figure 2. Superior NMI driver uses an oscillator (but requires power).

5) The application program has to be loaded explicitly with "N" and "L" commands.

When MODNMI.BIN is active, using the NMI trigger will sometimes *have no effect*. If, for instance, the program has called the operating system for keyboard input, execution will be looping inside the operating system. Whenever an NMI occurs under such conditions, the MODNMI code will correctly conclude that a breakpoint is inadvisable. This is *good*; breakpointing at such a location will result in a crash when an attempt is made to continue execution (as described previously). Programs do frequently call the operating system for input, but usually the programmer knows what's going on at this point, since typing a key or a return will cause program execution to continue. Usually, programs hang-up elsewhere.

TRIGGERING NMI

Different computers will have different approaches to NMI; on the Co-Power board in my Kaypro, the signal was connected directly to ground, so all I had to do was slice the trace and connect the little circuit depicted in Figure 1. Different computers would very likely require different values than those shown in Figure 1; it's pretty much a cut-and-try sort of thing.

For various complicated reasons having to do with other functions, I wanted my NMI trigger signal pulled high, even though the 8088 NMI input is high-to-low sensitive. Note, however, that the *transition* must occur to produce an NMI; the circuit shown in Figure 1 works because the 8088 remains reset as power comes on and NMI goes high, so it doesn't see the transition. When the momentary switch is pressed, NMI is brought low, but it immediately starts charging back to high through the 10k ohm

pull-up. The 390 pF capacitor is small, so this positive-going signal is reasonably speedy. The 10M ohm charge path around the capacitor ensures that a re-trigger will not occur for a while (i.e., it debounces the switch).

It is very important in any NMI scheme to avoid noisy re-triggering. The processor has no way of turning NMI off; a noisy switch connected directly to NMI could cause many triggers to occur so rapidly that available stack depth is exceeded (i.e., system crashes). A better circuit than that shown in Figure 1 would use active gates to implement a debouncing flip-flop.

Indeed, the best NMI triggering circuit would consist of a 555 timer or CMOS oscillator arranged to send a 1 KHz, or so, square wave to NMI when the switch is pressed. Figure 2 shows such a circuit. Programs can get hung-up in loops that include calling the operating system, filling the screen, say, with "@" signs, or smiling faces. With the circuit shown in Figure 1, one has to keep pressing the switch repeatedly to "catch" the program outside the operating system. The Figure 2 oscillator approach, on the other hand, will get the program "while you wait". The oscillator shouldn't be too fast or the problems described above with re-triggering will occur. (The oscillator's period should be long enough to allow execution of the NMI handler code in MODNMI.BIN. So long as this limit is observed, there will be enough time between one NMI and the next for the NMI handler to do the segment compares and exit with an IRET.)

IBM

I haven't tested MODNMI on an IBM computer; people are touchy about tearing down their computers and hooking things up to NMI. I did simulate an NMI (with an

INT 2) on a PC, and nothing caught on fire. The MODNMI code does include a test to avoid interrupting execution in a PC BIOS in high memory.

An examination of PC schematics shows a number of things hooked up to NMI: the 8087 numeric processor, parity check, and something called I/O CH CK (presumably input/output channel check). I don't know what the PC and SYMDEB do about these things, since I don't have a PC with NMI trigger at my disposal. If SYMDEB checks for parity errors or other NMI sources, then adding the MODNMI patch might suppress such events under the wrong conditions. I doubt it's a serious threat; certainly nothing compared to crashing at program re-entry. μ

James G. Owen designs equipment that sprays ink onto corrugated boxes for the Loveshaw Corporation on Long Island.

SCSI BUS INTERFACING continued

for writing is due to verifying every write operation. If each sector were not checked, there would be no way to recognize when an alternate sector should be allocated. Also, I am a cautious fellow and don't want to suffer the consequences of incorrectly written sectors, especially when it occurs in someone's file filled with treasured data.

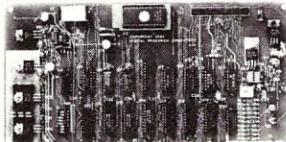
Note that the 4070 controller uses a *run length limited* encoding scheme which increases the capacity of an ST506 drive by a factor of 1.5. The bit rate is boosted to 7.5 Mb/s from 5 Mb/s. This has been very successful on the Priam V170 drives which use plated media. The plated media seems to be a requirement for RLL encoding on the ST506 type drives. Since the software for the 4000, which uses MFM encoding, and the 4070 are practically the same, upgrading from one to the other is simple. μ

DIGITAL RESEARCH COMPUTERS

(214) 225-2309

S100 EPROM PROGRAMMER

OUR NEWEST DESIGN, FOR FAST EFFICIENT PROGRAMMING OF THE MOST POPULAR EPROMS ON YOUR S100 MACHINE, COMES WITH MENU DRIVEN SOFTWARE THAT RUNS UNDER CP/M 2.2 (8 INCH). PC BOARD SET CONSISTS OF (S100) MAIN LOGIC BOARD REMOTE PROGRAMMING CARD AND SIX PERSONALITY MINI BOARDS FOR 2716, 2532, 2732, 2732A, 2764, AND 27128. SOLD AS BARE PC BOARD SET ONLY WITH FULL DOC. SOFTWARE FEATURES "FAST" PROGRAMMING ALGORITHM, FOR Z80 BASED SYSTEMS.



PC BOARD SET, FULL DOCUMENTATION, 8 IN. DISKETTE WITH SOFTWARE.

NEW! \$69.95

128K S100 STATIC RAM/EPROM BOARD

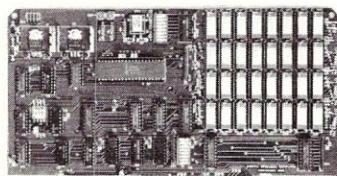
JUST OUT! USES POPULAR 8K X 8 STATIC RAMS (6264) OR 2764 EPROMS. FOR 8 OR 16 BIT DATA TRANSFERS! IEEE 696 STANDARD. LOW POWER. KITS ARE FULLY SOCKETED. FULL DOC AND SCHEMATICS INCLUDED. 24 BIT ADDRESSING.

NEW! \$59.95 **\$219.00** **\$139.00**
BARE PC BOARD 128K RAM KIT 128 EPROM KIT

256K S-100 SOLID STATE DISK SIMULATOR!

WE CALL THIS BOARD THE "LIGHT-SPEED-100" BECAUSE IT OFFERS AN ASTOUNDING INCREASE IN YOUR COMPUTER'S PERFORMANCE WHEN COMPARED TO A MECHANICAL FLOPPY DISK DRIVE.

PRICE CUT!



BLANK PCB
(WITH CP/M® 2.2
PATCHES AND INSTALL
PROGRAM ON DISKETTE)
\$49.95
(8203-1 INTEL \$29.95)

- * 256K on board, using +5V 64K DRAMS.
- * Uses new Intel 8203-1 LSI Memory Controller.
- * Requires only 4 Dip Switch Selectable I/O Ports.
- * Runs on 8080 or Z80 S100 machines.
- * Up to 8 LS-100 boards can be run together for 2 Meg. of On Line Solid State Disk Storage.
- * Provisions for Battery back-up.
- * Software to mate the LS-100 to your CP/M® 2.2 DOS is supplied.
- * The LS-100 provides an increase in speed of up to 7 to 10 times on Disk Intensive Software.
- * Compare our price! You could pay up to 3 times as much for similar boards.

\$129.00
(ADD \$50 FOR A&T)
#LS-100 (FULL 256K KIT)

ZRT-80

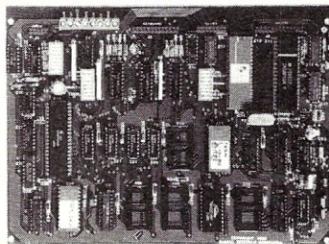
CRT TERMINAL BOARD!

A LOW COST Z-80 BASED SINGLE BOARD THAT ONLY NEEDS AN ASCII KEYBOARD, POWER SUPPLY, AND VIDEO MONITOR TO MAKE A COMPLETE CRT TERMINAL. USE AS A COMPUTER CONSOLE, OR WITH A MODEM FOR USE WITH ANY OF THE PHONE-LINE COMPUTER SERVICES.

FEATURES:

- * Uses a Z80A and 6845 CRT Controller for powerful video capabilities.
- * RS232 at 16 BAUD Rates from 75 to 19,200.
- * 24 x 80 standard format (60 Hz).
- * Optional formats from 24 x 80 (50 Hz) to 64 lines x 96 characters (60 Hz).
- * Higher density formats require up to 3 additional 2K x 8 6116 RAMS.
- * Uses N.S. INS 8250 BAUD Rate Gen. and USART combo IC.
- * 3 Terminal Emulation Modes which are Dip Switch selectable. These include the LSI-ADM3A, the Heath H-19, and the Beehive.
- * Composite or Split Video.
- * Any polarity of video or sync.
- * Inverse Video Capability.
- * Small Size: 6.5 x 9 inches.
- * Upper & lower case with descenders.
- * 7 x 9 Character Matrix.
- * Requires Par. ASCII keyboard.

FOR 8 IN. SOURCE DISK
(CP/M COMPATIBLE)
ADD \$10



\$89.95 A&T
#ZRT-80 ADD
\$50
(COMPLETE KIT, 2K VIDEO RAM)

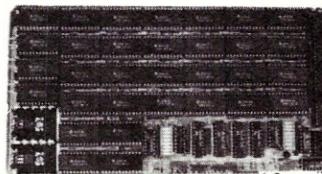
BLANK PCB WITH 2716
CHAR. ROM. 2732 MON. ROM.
\$49.95
SOURCE DISKETTE - ADD \$10
SET OF 2 CRYSTALS - ADD \$7.50

Digital Research Computers

P.O. BOX 381450 • DUNCANVILLE, TX 75138 • (214) 225-2309

64K S100 STATIC RAM

\$99.00
KIT



LOW POWER!

150 NS ADD \$10

BLANK PC BOARD
WITH DOCUMENTATION
\$49.95

SUPPORT ICs + CAPS
\$17.50

FULL SOCKET SET
\$14.50

FULLY SUPPORTS THE
NEW IEEE 696 S100
STANDARD
(AS PROPOSED)

ASSEMBLED AND
TESTED ADD \$50

FEATURES: PRICE CUT!

- * Uses new 2K x 8 (TMM 2016 or HM 6116) RAMs.
- * Fully supports IEEE 696 24 BIT Extended Addressing.
- * 64K draws only approximately 500 MA.
- * 200 NS RAMs are standard. (TOSHIBA makes TMM 2016 as fast as 100 NS. FOR YOUR HIGH SPEED APPLICATIONS.)
- * SUPPORTS PHANTOM (BOTH LOWER 32K AND ENTIRE BOARD).
- * 2716 EPROMs may be installed in any of top 48K.
- * Any of the top 8K (E000 H AND ABOVE) may be disabled to provide windows to eliminate any possible conflicts with your system monitor, disk controller, etc.
- * Perfect for small systems since BOTH RAM and EPROM may co-exist on the same board.
- * BOARD may be partially populated as 56K.

PANASONIC
Green Screen - Video Monitors
25 MHZ. TYPICAL BANDWIDTH!!!
Brand New In The Box! 9-Inch Screen
#K-904B1 (Chassis #Y08A) Open Frame Style

\$29.95

EA.

GROUP SPECIAL:
4 for \$99.00
WITH DATA & SCHEMATIC

(USA SHIPPING: \$3. PER UNIT. CANADA: \$7. PER UNIT)

COMPUTER MANUFACTURER'S EXCESS. STILL IN ORIGINAL PANASONIC BOXES. THE CRT TUBE ALONE WOULD COST MORE THAN OUR PRICE FOR THE COMPLETE UNIT. FOR SPLIT VIDEO (TTL INPUTS) OPERATION, NOT COMPOSITE VIDEO. OPERATES FROM 12VDC AT 1 AMP. VERTICAL INPUT IS 49 TO 61 HZ. HORIZONTAL INPUT: 15,750 HZ ± 500 HZ. RESOLUTION IS 800 LINES AT CENTER 650 LINES AT CORNERS.

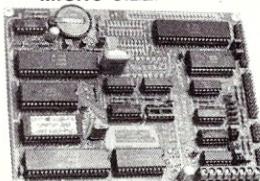
THE NEW 65/9028 VT ANSI VIDEO TERMINAL BOARD! * FROM LINGER ENTERPRISES *

A second generation, low cost, high performance, mini sized, single board for making your own RS232 Video Terminal. Use as a computer console or with a MODEM for hook up to any of the telephone-line computer services.

FEATURES:

- * Uses the new SMC 9028 Video Controller Chip coupled with a 6502A CPU.
- * RS-232 at 16 Baud Rates from 50 to 19,200.
- * On board printer port!
- * 24 X 80 format (50/60 Hz).
- * For 15,750 Hz (Horiz.) monitors.
- * 3 Terminal Modes: H-19, ADM3A, and ANSI X 3.64-1979
- * Wide and thin-line graphics.
- * White characters on black background or reversed.
- * Character Attributes: De-Inten, Inverse, Underline and Blank.
- * Low Power: 5VDC @ .7A, ± 12VDC @ 20MA.
- * Mini size: 6.5 X 5 inches.
- * Composite or split video.
- * 5 X 8 Dot Matrix characters (U/L case) with descenders.
- * Answer back capability.
- * Battery backed up status memory.
- * For ASCII parallel keyboard.

MICRO SIZE!



\$99.95

(Full Kit)

ADD \$40 FOR A&T

SOURCE DISKETTE:
PC/XT FORMAT
5 1/4 IN. \$15

OPTIONAL EPROM FOR
PC/XT STYLE SERIAL
KEYBOARD: \$15

TERMS: Add \$3.00 postage. Orders under \$15 add 75¢ handling. No C.O.D. We accept Visa and MasterCharge. Tex. Res. add 5-1/8% Tax. Foreign orders (except Canada) add 20% P & H. Orders over \$50 add 85¢ for insurance.

```

title modnmi.asm 5/9/85c jgo
page ,132 ;right margin is 65 w/elite type.

block group theseg,thedata ;everything will fit into
                           ;block segment.

theseg segment public 'code'

assume cs:block, ds:block, es:block, ss:block
org 0
zero:

org 100h
;1. get existing nmi vector, check that it's pointing to SYMDEB
;   NMI handler.
;2. Using segment obtained from vector, overlay SYMDEB sign-on
;   with our NMI handler.
;3. Replace existing nmi vector with vector to our handler.
signonat equ 137h ;in SYMDEB, signon's at CS:137h.
oldnmiat equ 1eb9h ;SYMDEB's nmi handler is at CS:1eb9h.

start: push es      ;save es.
       xor ax,ax    ;clear ax.
       mov es,ax    ;point at bottom of memory.
       les di,es:[8] ;get the nmi vector in es:di.
       cmp di,oldnmiat ;pointing right?
                           ;(user maybe forgot to do "/n" when
                           ;invoking SYMDEB.)
       jz domov      ;it's ok.

;no good.
       pop es        ;suppose we ought to restore es?
       mov dx,offset badstr
       mov ah,9        ;write string.
       int 21h        ;print the message.
       jmp zero       ;depart

badstr: db 'no NMI handler ("n"?).$'

domov: ;continue.

       mov di,signonat ;we'll put our code at the
                           ;sign-on.
       mov cx,nmiconcount ;how many bytes to move.
       mov si,offset ournmivec
       rep movsb        ;ok, handler in place.
       xor ax,ax
       mov es,ax        ;clear es again.
       mov es:[8],signonat ;change nmi vector to
                           ;point at new code.
       pop es
       jmp zero        ;back to SYMDEB.

;-----
;This is the improved NMI handler; it is embedded in this
;program, and we will load it over SYMDEB's sign-on
;message. This code will check that the program being NMied
;is in an appropriate region of memory -- so we don't
;attempt to trace MSDOS, SYMDEB itself, or the IBM BIOS, all
;of which will crash when re-entry is attempted.

ournmivec:
;This is supposed to go at cs:signonat in SYMDEB. Brave
;new high-tech world of 8088 assemblers doesn't provide
;grungy "phase" or "locate"-type commands, so
;theoffset equ ournmivec-signonat
;When we do an (8088-relative) jump to some SYMDEB
;location below, we'll jump to "location+theoffset".
       push ax
       push bp
       mov bp,sp      ;get stack pointer, because 8088
                           ;can't index sp.
       mov ax,[bp+6] ;get code segment of interrupted
                           ;program. At this point, stack looks like:
                           ;  high memory
                           ;    CS      stack+6
                           ;    IP      stack+4
                           ;    AX      stack+2
                           ;    BP      stack+0
       mov bp,cs      ;get our code segment.
;first of all, is interrupted code segment somewhere
;in high memory?
       cmp ah,0e0h
       jae onmi20    ;bad. Must be executing IBM
                           ;BIOS code.

```

QPARSER™

Translator W/riting System

THE PRODUCTIVITY TOOL FOR SOFTWARE DEVELOPERS

QPARSER assists you in writing:

- * Compilers * Translators * Interpreters *
- * Prototypes * Simulators * Syntax Checkers *
- * Data Converters * Assemblers *

QPARSER is a unique LALR(1) parser generator:

Generates complete source code for your application in C, Pascal, or another language of your choice; Extensive examples include a Pascal subset compiler, assembler, and simulator;

The widely used college text, *Compiler Construction: Theory & Practice*, from SRA Associates, was written by the author of QPARSER

Lauded by both industrial and university users

Available for: IBM PC,XT,AT; DEC VAX; HP 9816; MACINTOSH
(PC System \$400; Demo \$10; Educational/Site Licenses available)

"LEADERS IN SOFTWARE TOOLS"



1164 Hyde Ave., San Jose CA 95129

Toll-free Orders: (800) 538-9787; In CA: (408) 727-6671

```

;now let's see if SYMDEB's segment is greater or equal
;to interrupted code segment -- in which case we should
;return without breaking, because we're in MSDOS or
;SYMDEB itself.
       cmp bp,ax
onmi20: pop bp      ;restore regs.
       pop ax
       jae onmi40    ;bad. Either in IBM BIOS,
                           ;or MSDOS or SYMDEB itself.
;ok. proceed with SYMDEB's normal code.
       jmp oldnmiat+theoffset

;not ok. return harmlessly.
onmi40: iret

nmiconcount equ $-ournmivec

theseg ends

thedata segment word public 'data'
;actually there's no data in this program, but changing
;MASM mumbo-jumbo is treacherous, so I try to leave things
;the way they worked the last time.

thedata ends
end start ;this MUST be here to make com file.

```

μ

No source code for
your REL files?

REL/MAC

converts a REL file in the Microsoft™ M80 format to an 8080 or ZILOG™ Z80 source code MAC file with insertion of all public and external symbols.

- **REL/MAC** makes MAC source files
- **REL/MOD** lists library modules
- **REL/VUE** displays the bit stream
- **REL/PAK** includes all of the above
- 8080 REL/MAC demo disk \$10.00

REL/PAK for 8080 only \$99.95
REL/PAK for Z80 & 8080 \$134.95
on 8"SSSD disk for CP/M™ 2.2

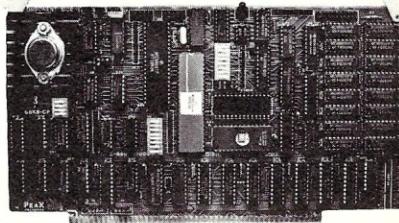
Send check, VISA, MC or C.O.D. to



MICROSMITH
COMPUTER TECHNOLOGY
PO. BOX 1473, ELKHART, IN 46515

1-800-622-4070

(Illinois only 1-800-942-7317)



68K8-CP

Expand Your System with a 68000 CoProcessor

Peak Electronics' 68K8-CP is a high performance 68000 software development package designed to easily integrate into your existing S-100 system. The package consists of the 68K8-CP coprocessor card, CP/M-68K, and a software toolkit that includes a UNIX V7 compatible floating point C compiler and a symbolic debugger.

Any system running CP/M®-2.2, CP/M-3.0 or CP/M-86 can be running CP/M-68K within minutes without any change in existing hardware or software. This card does not replace your current processor. All of the original system's devices (RAM, disks, and other peripherals) are immediately available to the user of CP/M-68K. All files can be accessed by whichever operating system is currently active. Control is transferred between operating systems with a simple one line command.

Features:

- Does not replace your current CPU card or software
- Includes CP/M-68K with UNIX® V7 compatible floating point C compiler and a symbolic debugger
- All developed C and Assembly code is fully relocatable and ROMable
- 8 or 10Mhz CPU with no wait state RAM
- 128K bytes of RAM expandable to 512K
- 2 serial and 1 parallel I/O ports
- IEEE-696-1983, S-100 Compatible
- 30 day money back guarantee
- 1 year parts and labor warranty

Complete Package: \$995.00
VISA or Master Card Accepted



P.O. Box 700112, San Jose, CA 95170-0112
(408) 253-5108

Up To Your Ears In Alligators?

If that sounds familiar, you need Write-Hand-Man™, the multi-function pop-up desktop organizer that works neatly with existing software for CP/M™ 2.2 and 3.0 systems. Write-Hand-Man eliminates that swamped feeling with tools that will get you organized. Write-Hand-Man comes with a 4-function, floating-point, 14 digit Calculator - Notepad - Two-week Appointment Book, File and Directory viewing - Phonebook with dialing - Cut and Paste - Key Redefinition - ASC II table. Even add your own applications.



Clear the swamp from your desktop.
Order Write-Hand-Man today. \$49.95

CA residents add 6.5% tax. Sorry, no credit cards or purchase orders. Specify: 8" or which 5" format CP/M 2.2 or 3.0 format

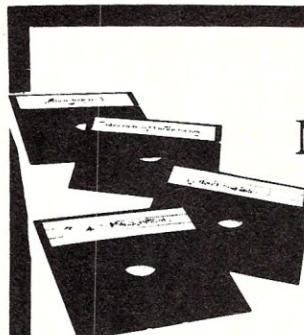
30 day guarantee

™Write-Hand-Man - Poor Person Software

™CP/M - Digital Research

Poor Person
Software

Dept. 204
3721 Starr King Circle
Palo Alto, CA 94306
(415) 493-3735



Micro/Systems Journal Software

Software appearing in
Micro/Systems Journal

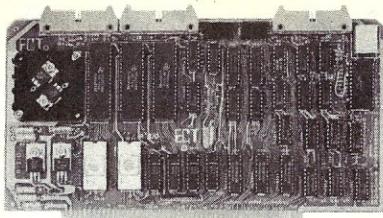
is released into the public domain via the SIG/M and PC/BLUE Public Domain Software Libraries. It may thus be copied at many computer club meetings and downloaded from many RBBS bulletin board systems.

The Software may also be obtained directly from us. We currently have two disks of software available:

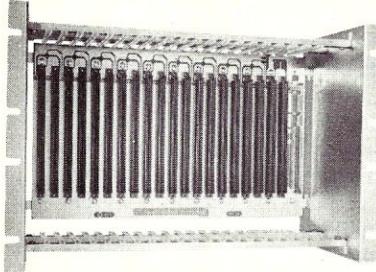
- Disk #1 - Software from 1985 issues
- Disk #2 - Software from 1986 issues

\$12/disk (includes shipping and handling). Outside North America add \$5 for shipping (via air). Ordering both disks deduct 10%. Send check (drawn on U.S. bank) or postal money order to: Micro/Systems Journal, Box 1192, Mountainside NJ 07042. Specify disk format.

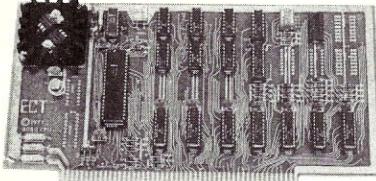
**CUSTOM
PRODUCTS**
DESIGN • LAYOUT
MANUFACTURING



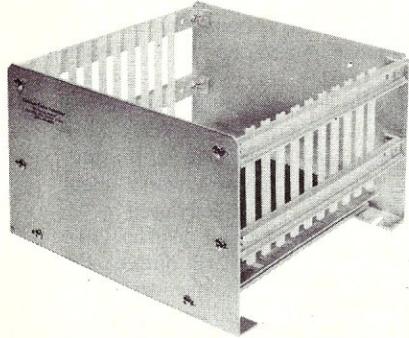
**S-100
PRODUCTS**



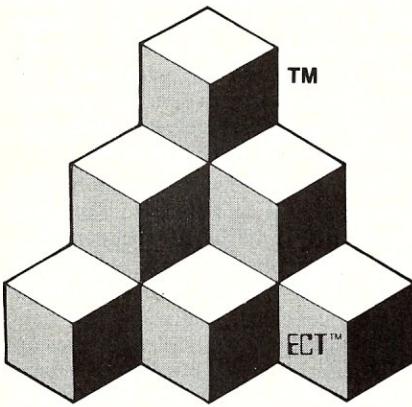
ECT-100-F
RACKMOUNT CARD CAGES



8080 CPU
CENTRAL PROCESSING UNITS

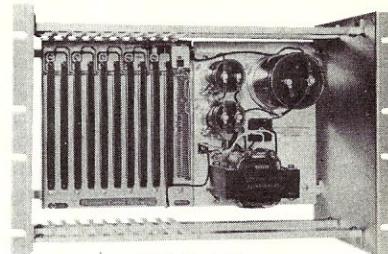


CCMB-10-F MIN
6,10 OR 20 SLOT CARD CAGES

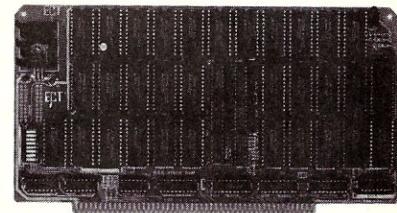


**BUILDING BLOCKS
FOR
MICROCOMPUTER SYSTEMS,
DEDICATED CONTROLLERS
AND TEST EQUIPMENT**

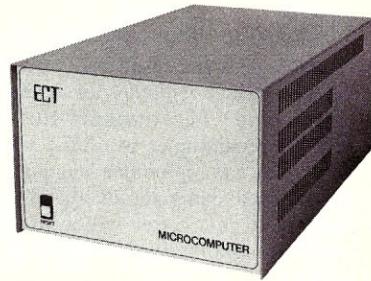
CARD CAGES, POWER SUPPLIES
MAINFRAMES, CPU'S, MEMORY
I/O, OEM VARIATIONS



RM-10
CARD CAGE & POWER SUPPLY



64K RAM
FULLY STATIC MEMORY

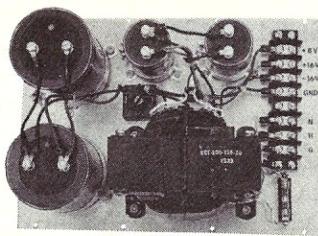


TT-10
TABLE TOP MAINFRAMES

ELECTRONIC CONTROL TECHNOLOGY, INC.

10 Cottage St., Berkeley Heights, NJ 07922 (201) 464-8086

**SPECIALIZING IN
QUALITY
MICRO COMPUTER
HARDWARE**



PS-30 A
POWER SUPPLIES

**MULTIBUS®
PRODUCTS**

MULTIBUS IS A TRADEMARK OF INTEL CORP.

SYSLIB, Z3LIB, & VLIB

by Richard Conn

The foundation of ZCPR3 and the Z System is a set of three software component libraries. This set of libraries consists of SYSLIB, a general-purpose library of over 200 routines, Z3LIB, a ZCPR3-specific library of over 100 routines, and VLIB, a ZCPR3-specific terminal interface library of over 10 routines. Each program in the ZCPR3 toolset is written in assembly language, and these libraries are used extensively throughout all the programs.

ASSEMBLY LANGUAGE AND HIGH-ORDER LANGUAGE PROGRAMMING

A purpose of high-order language programming is to raise the programmer to a higher level of abstraction, reducing the need to pay attention to implementation details and increasing his productivity. A simple equation in a high-order language like,

$$\text{COST} = \text{PRICE} * \text{QUANTITY}$$

expresses the meaning in terms a human can deal with easily, while a set of assembly language instructions like,

```
LOAD PRICE,B    ;Load register B with PRICE  
LOAD QUANTITY,A ;Load register A with QUANTITY  
MULT B,A        ;A = A*B  
STOR A,COST     ;Store product (in A) to COST
```

produces a similar result but is written in a language which is harder for the human to understand. The tradeoff lies in the reduced programmer overhead associated with using the high-order language versus the reduced computer overhead (smaller code size and faster execution speed) associated with using a lower-order language.

The users of the Z System typically do not care what language the Z System tools are written in; the users are mainly interested in programs that run quickly and reliably with little overhead. With this in mind, the 70+ programs supplied with the Z System are written in assembly language.

The hundreds of subroutines in these libraries can be grouped into two categories:

- 1) functions like sorting and multiplication, which are needed time and again. Rather than writing the same code over and over (reinventing the wheel), the code is written once in a general-purpose form, documented for future reference, and reused as the need arises.

- 2) functions like testing the wheel byte

The New ZCPR3 Libraries

or looking up a directory name to see what disk and user area it references, which require intimate knowledge of the internals of the Z System. Rather than remembering specific details, the programmer merely makes a subroutine call which tests the wheel byte or looks up a directory name for him and returns well-defined values that indicate the result of the operation.

The tradeoff between assembly language programming with and without the libraries is that the library routines, as a result of their general-purpose nature, are slightly less efficient than customized assembly language routines which perform similar functions. The result of using the libraries in assembly language programming is typically far more efficient than using a high-order language, and, if the need exists due to time or memory constraints, the programmer can take the source code of the library routines and customize it to make the code for his application more efficient.

I have been a fan of programming in high-order languages, such as Ada and C for many years, and a primary goal in creating the ZCPR3 libraries was to receive the benefits associated with the use of high-order languages (more structured code, improved maintainability, improved readability, and reduced programmer effort) while simultaneously realizing the speed and efficiency of assembly language. I believe this is a fundamental requirement in eight-bit systems. In more sophisticated systems like a 32-bit VAX 11/780 running UNIX, the advantage of using assembly language over C is negligible, and I don't consider using any languages other than C or Ada in these environments.

THE SYSLIB LIBRARY

SYSLIB is the general purpose library, and it can be used to create programs for use under normal CP/M or under a Z System. The current version of SYSLIB generates code for a Z80 or instruction-set compatible (eg. HD64180) microprocessor. Earlier versions of SYSLIB generated code for an 8080 or instruction-set

compatible microprocessor. SYSLIB may be used by the MACRO-80/LINK-80 assembler and linker, the ZAS/ZLINK assembler and linker, or any other assembler and linker that can work with object files in the Microsoft REL format. SYSLIB Version 3.6 (the most recent version) is written in the ZAS assembler language and can only be assembled by the ZAS assembler (SYSLIB.REL is provided in the distribution, so it is not necessary to be able to assemble SYSLIB in order to use it). SYSLIB's routines fit into nine categories:

- 1) character input/output
- 2) string input/output
- 3) numeric input/output
- 4) file manipulation
- 5) directory manipulation
- 6) user area and disk control
- 7) branching
- 8) mathematical functions
- 9) collections of reusable utilities

The character, string, and numeric input and output routines provide functions such as obtaining a character from the console, outputting a string to the console, inputting a line from the console with editing (backspace, etc.), outputting an 8- or 16-bit number in a variety of forms, and converting the ASCII text representation of a number into a 16-bit number. As with all SYSLIB routines, unnecessary side effects have been eliminated, so the programmer does not have to worry about effects on registers not directly used by the routines. In Listing 1, for instance, the PRINT routine in SYSLIB outputs the string *The value of HL is* to the console and does not have any effect on the value in HL itself. A second example, showing how numeric input, output, and processing can be done, is presented in Listing 2.

In contrast, similar programs were written using the Turbo Pascal high-order language (Listings 1P and 2P). The two pairs of programs (Listings 1 and 1P, 2 and 2P) exhibit several common characteristics: they produce similar results, they are relatively easy to read, and they are small and efficient. The sizes of their object (COM) files, however, are quite different: Listing 1 assembled into a 3 record file and Listing 1P compiled into a 65 record file, while Listing 2 assembled into a 7 record file and Listing 2P compiled into a 66 record file. The assembler programs were on the order of 1K bytes in size, while the Turbo Pascal programs were on the order of 9K bytes in size.

The file manipulation routines of SYSLIB provide functions for working with disk files, such as:

- 1) appending to the end of an existing file
- 2) creating, opening, closing, deleting, and renaming files
- 3) testing for the existence of files
- 4) computing the sizes of files
- 5) reading from and writing to files sequentially or randomly
- 6) reading from and writing to files a byte at a time
- 7) scanning and extracting components from LBR files

The directory manipulation routines provide functions for:

- 1) loading a disk directory into memory
- 2) sorting a disk directory in memory
- 3) selecting file entries from a disk directory
- 4) computing the sizes of files from their entries
- 5) computing the amount of free space left on disk

The user area and disk control, branching, and mathematical routines include functions for:

- 1) logging into a directory
- 2) saving your current directory location and returning to it later
- 3) implementing case, computed goto, and arithmetic IF branching
- 4) performing arithmetic, logical, rotate and shift, random number generation, and CRC calculation operations

Finally, the reusable utilities include functions for:

- 1) memory allocation
- 2) parsing, including a UNIX-style ARGC/ARGV parser
- 3) in-memory sorting
- 4) string and vector comparison
- 5) determining the end of code

THE Z3LIB LIBRARY

Z3LIB is a ZCPR3-specific library which provides a variety of routines useful to the assembly language programmer who is writing on a utility to run under ZCPR3. The current version of Z3LIB generates code for a Z80 or instruction-set compatible (eg, HD64180) microprocessor. Z3LIB may be used by the MACRO-80/LINK-80 assembler and linker, the ZAS/ZLINK assembler and linker, or any other assembler and linker that can work with object files in the Microsoft REL format. Z3LIB Version 1.3 (the most recent version) is written in the ZAS assembler language and can only be assembled by the ZAS assembler (Z3LIB.REL is provided in the distribution, so it is not necessary to be able to assemble Z3LIB in order to use it).

Listing 3 illustrates some of the routines available in Z3LIB. Note that the routine Z3INIT must be called before any other Z3LIB routine. The address of the



Z SETS YOU FREE!

Z Operating System, an 8-bit OS that flies! Optimized HD64180/Z80 assembly language code — full software development system with proven linkable libraries of productive subroutines — relocating [ROM and RAM] macro assembler, linker, librarian, cross-reference table generator, debuggers, translators and disassemblers — ready to free you!

High performance and flexibility! Productivity results from dynamically customized OS environments, matching operator, tasks and machine.

Real-time control kernel option allows quick software development for industrial control applications, other tools and utilities for office desk-top personal computing functions, local area networks to Ethernet, AppleTalk, Omnitel, ArcNet, PC-Net (Sytek) — from micro to mainframe command, control and communications. Distributed processing application programs are easily developed.

- Extreme organizational flexibility, each directory another environment
- Multiple Commands per line
- Aliases (complex series of commands known by simple names) with variable passing
- Named Directories with absolute password security
- Full-screen command line editing with previous command recall and execution
- Shells and Menu Generators, with shell variables
- Command-file search Paths, dynamically alterable
- Screen-oriented file manipulation and automatic archiving and backup
- 512 megabyte file sizes, 8 gigabyte disks handled
- Auto disk reset when changing floppies
- TCAP database handles characteristics of over 50 computers and terminals, more easily added
- Tree-structured online help and documentation subsystem
- 76 syntax-compatible support utilities

Your missing link has been found — Z! Now fly with eagles! Fast response, efficient resource utilization, link to rest of computing world — shop floor to executive suite, micro to corporate mainframe. Call 415/948-3820 for literature.



Echelon, Inc.

101 First Street • Suite 427 • Los Altos, CA 94022 • 415/948-3820

ZCPR3 Environment Descriptor (a data buffer which contains details on a specific ZCPR3 system) is passed to Z3INIT in the HL register pair, and Z3INIT makes this address available to all routines in Z3LIB. Z3LIB makes it quite easy to determine useful information on the particular ZCPR3 system under which the program is running. The program in Listing 3 accesses the speed of the processor and other attributes of the ZCPR3 System.

The functions provided by Z3LIB include:

- 1) modification and examination of all aspects of the ZCPR3 environment
- 2) manipulation of the IF/ELSE/FI flow control facility

- 3) manipulation of the ZEX memory-resident command file processor
- 4) named directory and user/disk conversion
- 5) command line and token parsing
- 6) timing routines
- 7) path search for files

THE VLIB LIBRARY

VLIB is a ZCPR3-specific library that serves to provide the user with routines for screen manipulation through the ZCPR3 TCAP (terminal capabilities data base). The current version of VLIB is 1.1. By using VLIB, the ZCPR3 system utility programmer can write transportable, screen-oriented software which runs on any of the 60+ terminals supported by the

Instant-C™ The Fastest Interpreter for C

Runs your programs 50 to 500 times faster than any other C language interpreter.

Any C interpreter can save you compile and link time when developing your programs. But only **Instant-C** saves your time by running your program at compiled-code speed.

Fastest Development. A program that runs in one second when compiled with an optimizing compiler runs in two or three seconds with **Instant-C**. Other interpreters will run the same program in two minutes. Or even ten minutes. Don't trade slow compiling and linking for slow testing and debugging. *Only Instant-C will let you edit, test, and debug at the fastest possible speeds.*

Fastest Testing. **Instant-C** immediately executes any C expression, statement, or function call, and display the results. Learn C, or test your programs faster than ever before.

Fastest Debugging. **Instant-C** gives you the best source-level debugger for C. Single-step by source statement, or set any number of conditional breakpoints throughout your program. Errors always show the source statements involved. Errors always show the source statements involved. Once you find the problem, test the correction in seconds.

Fastest Programming. **Instant-C** can directly generate executable files, supports full K & R standard C, comes with complete library source, and works under PC-DOS, MS-DOS, or CP/M-86. *Instant-C gives you working, well-tested programs faster than any other programming tool.* Satisfaction guaranteed, or your money back in first 31 days. **Instant-C** is \$495.

Rational
Systems, Inc.

P.O. Box 480
Natick, MA 01760
(617) 653-6194

ZCPR3 TCAP. The functions provided by VLIB include:

- 1) clear screen
- 2) cursor positioning
- 3) erase to end of line
- 4) highlighting

Listing 4 illustrates some of the VLIB routines. Note that the routine Z3VINIT is used to provide the address of the environment descriptor to the VLIB routines, and Z3VINIT also provides this information to Z3LIB routines (Z3INIT is not needed if Z3VINIT is used).

THE LIBRARIES & THE NEW ZCPR 3.3

The first step in creating the new

ZCPR 3.3 system involved the redesign of the environment of the system and an extensive review and redesign of the libraries. The first redesign of the libraries is complete, and the source code, online documentation, and REL files which comprise the three ZCPR3 libraries, SYSLIB, Z3LIB, and VLIB, have been released through Echelon and SIG/M (Volumes 261-264 of SIG/M). Now that the ZCPR 3.3 command processor is finished, a second redesign of the libraries is in order. The internals of the library routines, especially those routines in Z3LIB, will be changed extensively, while the interfaces (the input and output parameters especially) will stay the same. Once the new

Z3LIB is complete, all that needs to be done to the 70+ programs of the Z System is to reassemble them using the new Z3LIB, and they will continue to function under the new ZCPR 3.3 as they did under the old ZCPR 3.0.

THE LIBRARIES DOCUMENTATION

The libraries are fully documented in a loose-leaf book, *ZCPR3: The Libraries*, published by Echelon, Inc. In addition, the distribution of the libraries includes a number of help (HLP) files which can be read and displayed by the HELP and HELPPR programs of the ZCPR3 toolset. Both the book and the HLP files document every routine in the libraries, giving the name of the routine, its function, its input and output parameters, and other information of interest to the programmer. As the libraries change, the insert pages for *ZCPR3: The Libraries* will be sent out by Echelon. The price of *ZCPR3: The Libraries* is \$30.95, and it can be ordered from Echelon, Inc. (see ad in this issue) or through Micro/Systems Journal (see Book Mart section).

ZCPR3: The Libraries is over 300 pages in size, and is divided into five sections: an introduction, SYSLIB, Z3LIB, VLIB, and a user's guide. All routines in the libraries are described, and the user's guide provides style of use with many examples. Three appendices include nine sample programs, a listing of all files, and update notices. A comprehensive 5-page table of contents and an extensive 12-page index provide efficient access to the information.

GETTING THE LIBRARIES

The source code, REL files, and HLP files of libraries can be obtained from three sources:

- 1) Echelon, Inc. (\$69.00)
- 2) SIG/M (Volumes 261-264)
- 3) RCP/M and Z-Node Electronic Bulletin Boards

Richard Conn is both a computer professional and a computer "fan." Rick holds a Master of Science degree in Computer Science from the University of Illinois at Urbana, and has been working in the computer field since 1978 on items ranging from embedded computers in satellite systems to the design and implementation of microcomputer-based operating systems. He is probably best known as the creator of SYSLIB, Z3LIB, VLIB, ZCPR2, and ZCPR3. Rick currently runs the DoD Ada Software Repository on the Defense Data Network and continues to work on the evolution of ZCPR3.

Trademark acknowledgements: VAX - Digital Equipment Corporation, ZCPR3, Z System, ZAS, and ZLINK - Echelon, Inc., MACRO-80 and LINK-80 - Microsoft, Inc., Turbo Pascal - Borland International, Inc., Z80 - Zilog Corp., HD64180 - Hitachi, Ltd.

Listing 1

```

ext print,phldc ; Reference SYSLIB Routines
;
ld hl,340 ; Store a value into HL
call print ; Print a string
db 'The value of HL is ',0
call phldc ; Print HL as decimal characters
           ; " 340" will be printed
ret
;
end

```

Listing 2

```

ext print,phldc ; Reference SYSLIB Routines
ext bbline,eval10,crif
ext mulhd
;
call print ; Print a prompt
db 'Input the first number: ',0
call bbline ; Input into a buffer with editing
           ; At this point, HL contains the
           ; address of the internal buffer
           ; used by BBLINE
call crlf ; Output a new line to the console
call eval10 ; Convert the number pointed to by
           ; HL into a value in DE
ld (value1),de ; Store the first value
;
call print ; Prompt for the second number
db 'Input the second number: ',0
call bbline ; Same as before
call crlf
call eval10 ; DE now contains the second number
           ; input by the user
;
ld hl,(value1) ; Get the first value into HL
call mulhd ; HL = HL * DE
;
call print
db 'The product of these numbers is ',0
call phldc ; Print the product
ret
;
value1: ds 2 ; Buffer for storing VALUE1
;
end

```

Listing 1P

```

program 11;
var
  value1 : integer;
begin
  value1 := 340;
  writeln ('The value of VALUE1 is ', value1);
end.

```

Listing 2P

```

program 12;
var
  value1, value2 : integer;
begin
  write ('Input the first number: ');
  readln (value1);
  write ('Input the second number: ');
  readln (value2);
  value1 := value1 * value2;
  writeln ('The product of these numbers is ', value1);
end.

```

Listing 3

```

ext z3init ; Z3LIB References
ext getspeed ; Returns the processor speed
ext getwhl ; Returns the wheel byte
ext getquiet ; Returns the quiet flag
;
ext print,crlf ; SYSLIB References
ext padc
ld hl,0f400h ; Address of ZCPR3 Environment
call z3init ; Descriptor
call print
db 'The speed of your processor is ',0
call getspeed ; Get speed in Register A
call padc ; Print Register A as decimal
call print
db 'MHz',0
call crlf
call print

```

TURBO PASCAL GENERATOR

THE GTP PROFESSIONAL MODEL

Generate error-free Turbo Pascal source code for:

**DATA BASES MENUS
SCREENS REPORTS**

Build complete, working programs in minutes!

FEATURES

- Indexed Data Bases
- Multiple Screens
- Automatic Updating
- Built-in Edits
- Retrieval Facility
- Automatic File Build
- Quick Screen Handling
- Speedy DB Access
- Context Sensitive HELP
- Full Keyboard Support

EASY

1. Paint-the-screen
2. Define fields & calculations
3. Generate, compile & RUN

FLEXIBLE

- 100% Turbo Pascal
- Modify in Pascal under Turbo's Editor
- Extensions are easy with 130 page Programmer Reference manual
- 100 documented source code routines included

NOT Copy Protected NO Royalty Fees NO Run-time Library Required

Requires: IBM PC (100% compatible) 256K RAM 2 Disk Drives
PC-DOS 2.0+ Turbo Pascal 3.0 from Borland International

PRICE \$200.00

CALL FOR QUANTITY DISCOUNTS

VISA/MC - Check - Money Order - No COD or Purchase Orders
Texas Residents: add \$12.25 Sales Tax
Outside US & Canada: add \$25.00 Air Postage and make payment by
credit card or money order in US Funds.



ALLEN, EMERSON & FRANKLIN

P.O. Box 928
Katy, TX 77492
(713) 391-8570

```

db 'The Wheel Byte is ',0
call getwhl ; Get wheel byte
call pronoff ; Print On/Off
call crlf
call print
db 'The Quiet Flag is ',0
call getquiet ; Get quiet flag
call pronoff ; Print On/Off
ret
;
```

```

pronoff:
or a ; if A=0, print OFF
jr z,proff
call print
db 'On',0
ret
;
proff:
call print
db 'Off',0
;
end

```

Listing 4

```

ext z3vinit ; Reference VLIB Routines
ext tinit,dinit ; Terminal initialization/deinit
ext stdndout,stdndend ; Begin/End Highlighting
ext cls,at ; Clear screen, position cursor
;
ext print ; From SYSLIB
;
ld hl,0f400h ; Address of the ZCPR3 Environment
call z3vinit ; Descriptor
call tinit ; Initialize the user's terminal
call cls ; Clear the screen
call at ; Position on the user's screen
db 10,25 ; at Row 10, Column 25
call stdndout ; Begin highlighting
call print ; Print a string
db 'Hello, World',0
call stdndend ; End highlighting (normal video)
call at ; Position at bottom left corner
db 23,1
call dinit ; Deinitialize terminal
ret
;
end

```



PRINTERS & BUFFERS

M-1109 100-25 CPS, P&S, TRACTOR \$203
 HR-35 36CPS DAISEY, WIDE CARR .. \$729
 HR-20 20 CPS \$349

CITIZEN MSP-10 .. \$250 MPS-15 .. \$349
 CITIZEN MSP-20 .. \$315 MPS-25 .. \$485
 CORDATA LP-300 LASER WITH WORDSTAR 2000 & FORMS-EASY ... \$2175
 HANZON BUFFER 64K-256K, S-S, S-P, P-S, P-P \$265
 HANZON LP-3000+ LASER W/ HP GL PLOTTER COMMANDS..... CALL
 STAR MICRONICS SR-15 \$595
 TOSHIBA P321 PARA & SERIAL \$449
 TOSHIBA P341 \$769
 TOSHIBA P351 PARA & SERIAL \$969
 OKIDATA 292 COLOR PRINTER \$459
 OKIDATA 293 CLR W/ TRC. \$599

XEROX Diablo

D25 \$484 635 \$1049
 D80-IF \$2049 34LQ \$989
 4045 LASER W/COPIER,512K RAM\$4795

PERSONAL SPEECH SYSTEM \$295

POWER SYSTEMS & ACCESS

COMPUTER ACCESSORIES

ALL CABLES Call for Lg Discount!
 S4 4 CIRCUIT SURGE SUPPRESSOR .. \$26
 S10 6 CIRCUIT SURGE SUPPRESSOR \$29
 P15 5 CIRCUIT MONITOR BASE WITH MODEM PROTECTION \$98
 U1000 POWER SAVER, (TOPAZ) 1000W U.P.S. \$729

SB-1000 WATT W/ BATT 30-60 MIN. .. \$659
 BC-1000 WATT WITH SELF CONTAINED BATTERY, 15-30 MIN. \$889

SAFE (SAFT) SPS1000WATT SINE WAVE \$989

MODEMS

PROMODEM 1200 \$259
 PROMODEM 1200G \$179; 2400G \$339
 PROMODEM 300C APPLE IIC \$79

MT224EH MNP ERROR CORREC \$529
 MT224ER RACK MOUNT W/ EDC \$495

CTS 212HC HAYES COMPATIBLE \$149
 CTS 224ADH HAYES COMPATIBLE ... \$295

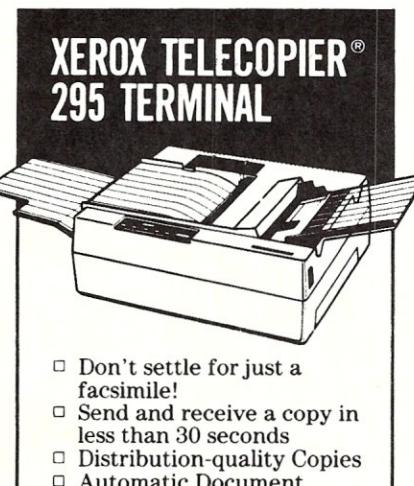
HAYES -ALL MODEMS CALL

MONITORS & TERMINALS

NEC MULTISYNC \$515
 TECMAR RGB HI-RES ZVM-1360 \$498
 ZENITH ZVM-1240 (IBM) FLAT SCRN ... 149
 PRINCETON GRAPHICS CALL

CM-1360 640x200 RGB, 13", G/A SW. \$359
 CM-1380 EGA 640x350 RES \$479
 1422A DUAL FREQ, TILT/SWIVEL, 14", GREEN/AMBER \$139

LINK 125 EMUL WY-50 ETC., HI-RES 14" G/A, 6 SCROLL RATES, IBM SELEC, K.B., 1 YEAR WARRANTY \$385
 PC-TERM EMUL WY-50, TVI 925, IBM-AT KB, GOOD FOR PC-SLV BDS \$419
 WYSE 60 \$475
 LIBERTY FREEDOM ONE \$395



- Don't settle for just a facsimile!
- Send and receive a copy in less than 30 seconds
- Distribution-quality Copies
- Automatic Document Feeder: 30 originals
- 328' Paper Roll
- RS-232-C Port Capabilities
- CCITT Groups 3, 2 and 1
- Compatibility and more!

\$2139

XEROX*, TELECOPIER* and the identifying numbers herein are trademarks of XEROX CORPORATION.

SOFTWARE

We sell all well known brands. Order correctly - OPENED SOFTWARE IS NOT RETURNABLE!

NEWSTAR NEWWORD 2 \$169
 NEWSTAR NEWWORD 3 \$269
 BD SOFTWARE C COMPILER 8".... \$95
 MICROSOFT * ASHTON-TATE - 40%
 MICROPRO * MULTIMATE * PFS - 45%
 BORLAND TURBO PASCAL, ETC ... - 45%
 MOST OTHER PC PROGRAMS ... - 45%

NETWORKING

CALL FOR DEALER PRICING, INTEGRATION, AND/OR CONSULTATION.

COMPUPRO NET 100 \$369

PC-SLAVE/16 1MB, 8MHZ, V20 \$595
 RTNX MSDOS 2.1 \$109
 ATNX MSDOS 3.1 \$189

LAN-PC ARCNET FOR PC'S \$349
 WS-286 8MHZ 80286, 5 SLOTS, ARCNET I/F, 640K, SCSI, AT TYPE ENCLOSURE & POWER SUPPLY \$1398

ADVANCED NETWARE STARTER KITS SOFTWARE, KEYCARD, 2 NICs, CABLES INCLUDED FROM \$1795

USER NET SW / 86 & KEYCARD \$895

CABINETS

MPS 10005 6 SLOTS 3HH DRVS	\$395
MPS 5401 HARD DISK FH OR 2HH	\$195
JMR 1H5 PC-STYLE FH OR 2 HH HARD DISK DRIVE	\$185
INTEGRAND 2915DUAL HORIZ. FH HARD DISK DRIVE	\$199
INTEGRAND 800AD/1515 SLOTS 2 HARD DISK (5")	\$440
INTEGRAND LASER 3310 4 SLOTS, 2 HARD DISK (5")	\$387
PARA DYNAMICS 3020D 20 SLOT DESK TOP	\$595
PARA DYNAMICS 2300-G3 5" HD & 2x8 FLOPPY	\$339
PARA DYNAMICS 5820-S 58 AMPS, 20 SLOTS, MANY DRIVES	\$1795

HARD DISK SYSTEMS

40MB Q540 W/ DISK 3, CABINET, CABLES, ETC. FOR COMPUPRO	\$1625
80MB MC1325 FOR COMPUPRO ..	\$2139
COMPUPRO DRIVE ENCL, DISK 3, QUANTUM Q540, 5" & 8" FLPYS	\$2498

DRIVES

SEAGATE ST-225 20MB, 65mSEC	\$319
QUANTUM Q540 40MB, 45mSEC	\$995
QUANTUM Q540 REFURB 1 YR	\$839
SEAGATE 4051 50MB, 40mSEC	\$825
MICROPOLIS 1325 85MB, 28mSEC	\$1395
MAXTOR XT-1140 140MB, 30mSEC	\$3195
ALLOY 17.7MB TAPE SUBSYS	\$1659
IRWIN 10MB TAPE (CDOS4.1/CPRO)	\$369



All merchandise new. Advertised prices are cash prepaid only. AM. Express - add 5%. MC, Visa & P.O.'s from qualified firms - add 3%. Wires, COD's (\$5 min. fee) with Cashiers Check/MO & APO's accepted. Shipping: minimum \$4 first 3 lbs. Tax: AZ RES ONLY add 6 1/2% sales tax. All returns subject to 20% restocking fee or credit towards future purchases. All prices subject to change without notice.

When Our Ad Says "Call", We Will Not Be Undersold!

S-100™

S-100 BUS PRODUCTS

CompuPro

We will integrate a system of some or all CompuPro components
"HAVE SPECS WILL QUOTE!"

SYSTEM 816/C-2	\$7279
CPU286 8MHZ, 3 CYCLE	\$509
CPU286 8MHZ, 2 CYCLE	\$655
CPUZ 8MHZ Z80	\$199
CPU 8085/88 6/10MHZ	\$255
CPU 8086 10MHZ	\$288
SPUZ 256K, 8MHZ, UP TO 4 USERS	\$279
SP186 10MHZ, 512K RAM SLAVE	\$475
MDRIVE/H 512K	\$363
MDRIVE/H 2MB	\$800
RAM 23 64K \$180 .. RAM 23 128K ..	\$255
RAM 22 256K STATIC ..	\$435
RAM 24 1MB STATIC ON "ONE BD" ..	\$2195
RAM 16 64K STATIC RUNS W/ CPU286	\$99
INTERFACER 3 8 SERIAL ..	\$435
INTERFACER 4 3 SERIAL / 1 PARAL ..	\$329
DISK 1A 5" & 8" FLPY CTRL ..	\$435
DISK 3 ST-506 H.D. CTRL ..	\$508
ENCLOSURE/2-DESK 21 SLOT ..	\$729
ENCLOSURE/2-RACK 21 SLOT ..	\$800
DRIVE ENCLOSURE-DESK ..	\$508
DRIVE ENCLOSURE-RACK ..	\$655
PC VIDEO BOARD ..	\$362
NET 100 ..	\$362
HYBRID HUB \$255 / PASSIVE HUB \$47	
SYSTEM SUPPORT 1 ..	\$292
CONCURRENT DOS 4.1D ..	\$495
CONCURRENT DOS 4.1E ..	\$635
CPM/816 .. \$255	CP/M 80 .. \$182

MACROTECH INTERNATIONAL CORP. 

MI-286 80286, 8MHZ, 2-CYCL & Z80H, A "DROP-IN" REPLACEMENT FOR	
CPU 8085/88 ..	\$666
MSR-II 1MB DRAM ..	\$808
MSR-512 512K DRAM ..	\$555
ST-II 1MB STATIC RAM ON "1 BD" ..	\$1999
ADIT-8 INTELLIGENT 8 PORT SERIAL	\$777
AMADIT-8 FOR ALPHA MICRO SYS. ..	\$959

LDP

Let us integrate a system per your specs! Ask about our stylish 6 slot 1.5 drive cabinet.

THUNDER+ 8MHZ 80186, 512K RAM, 2S, P, FLPY CTRL, CDOS 4.1 -LITERALLY A SYSTEM ON A BOARD ..	\$895
THUNDER+ 10MHZ 80186, 1MB RAM, ETC. AS ABOVE ..	\$1095
LIGHTNING 286 8MHZ CPU ..	\$819
LIGHTNING 286 10MHZ 80286 CPU ..	\$897
CONTROL-IT-ALL H.D.&FLPY CTRL	\$485
HAZITALL 2 S, 2 P, CLOCK ..	\$245
MEGARAM 512K DRAM ..	\$445
MEGARAM 1MB DRAM ..	\$595
MEGA-S-RAM 16K-1MB STATIC RAM	\$298
OCTAPORT 8 PORT SERIAL ..	\$298
NV-DISK SOLID STATE 2MB DISK ..	\$745
CDOS 4.1 MULTI-USER ..	\$350
COLOR MAGIC S-100 BUS PC COLOR GRAPHICS EMULATOR FOR BUS.....	\$445

S-100 DIV./696 CORP.
14455 NORTH 79th ST.
SCOTTSDALE, AZ 85260

S-100 BUS PRODUCTS

**InterContinental
Micro**

Registered ICM dealers get preferred prices. Consultants, we will integrate your system or provide all components.

THIS MONTH'S SPECIAL:

4 USER 6MHZ, 8 BIT SYSTEM W/ 20MB HARD DISK, 5"FLPY (20MB TAPE DRIVE OPTIONAL), & 4 ADDITIONAL SLOTS FOR FUTURE EXPANSION	\$3595
CPZ-48006 MASTER: 6MHZ Z80, 64K, 2S, 2P, FLPY, 8&16 BIT MEMORY OR I/O MAPPED APPLICATIONS	\$733
CPS-B8A SLAVE: 8MHZ Z80, 128K, P, 2S ..	\$439
CPS-Q6A 6MHZ, 128K, 4 SLAVES ON ONE BOARD ..	\$1398
CPZ-186 MASTER: 8MHZ, 1MB RAM, 2S, WILL RUN 8 AND/OR 16 BIT APPLICATIONS AND SLAVES ..	\$1119
CPS-16F SLAVE: 10MHZ 8086, 1MB RAM, 2S, P ..	\$769

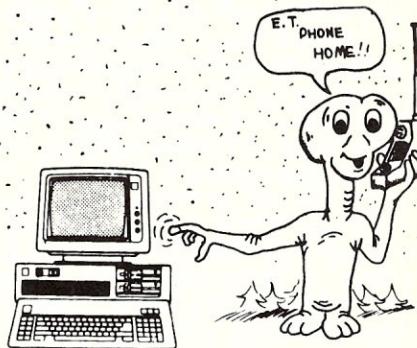
CPS/MS1000-PKG: CPS-16F, PC-DOS 2.11, TURBODOS/PC, ICMS PC-DOS UTILITY FOR TURBODOS & 14" TERMINAL THAT WILL RUN ALL PC MONOCHROME PROGRAMS... IE: LOTUS 1-2-3, SIDEKICK, WORDSTAR, MULTIMATE, dBASE III, ETC. THAT DO NOT MAKE DIRECT PC HDWRE CALLS ..	\$1398
CPS-186 SLAVE 10MHZ 1MB RAM \$1049	
CPS-216 DUAL SLAVE: 8MHZ 8086 & 512K ..	\$1189
OMTI-5300 HD & TAPE CTRL CONNECTS WITH SCSI PORT ON MASTER, MOUNTS ON DRIVE ..	\$489
LANS-100 ARCNET CONTROLLER ..	\$349
LAN-PC ARCNET CONTROLLER (WORKS WITH NOVELL) ..	\$349
WS-286 SERVER 8MHZ, 5 SLOTS, 200W, 640K, ARCNET, SCSI & SASI CONTROLLERS ON MOTHER BD ..	\$1609
TURBODOS MUTD-E FOR 8 OR 16 BIT MASTER & SLAVES ..	\$525
TURBODOS MUTD-I FOR ANY COMBO, 8 & 16 BIT ..	\$875
NOV-86 NOVELL NETWARE 86 SW	\$1119

VME BOARDS

DUAL

VMPU-32 PROCESSOR 68020 16 MHZ, 1MB RAM, CLOCK, BATT, PAGED MEMORY MGMT ..	\$4200
VSMD CACHING SMD 3 DRV.CTRL ..	\$2392
VMEM 2MB, 85mSEC ACCESS ..	\$3196
V9TRK 9-TRK TAPE CTRL, 6250 BPI ..	\$999
VIOP I/O CPU, 10MHZ 6800, 512K ..	\$1197
IO-S8 8 SERIAL UP TO 84.4K BAUD ..	\$397
SCM SYS CONTROL MODULE, 16MHZ	\$397
VMBD-12 12 SLOT BACKPLANE ..	\$669

EVERYONE'S PHONING HOME
ABOUT THIS ADVANCED "AT"
COMPATIBLE SYSTEM



"ET-286" BASIC SYSTEM

12MHZ ONE WAIT STATE OF 10MHZ NO
WAIT STATE SPECIAL ORDER - ALLOW 5
WEEKS. 10 & 6 MHZ 80286 CPU ONE WAIT
STATE, 1 - 4MB RAM ON MOTHER BOARD
ALSO ON MOTHERBOARD:

- * 6 AT COMPATIBLE SLOTS
- * 2 XT COMPATIBLE SLOTS
- * 3 RS232 OR RS422 SWITCHABLE SERIAL
PORTS
- * 2 CENTRONICS PARALLEL PORTS
- * CLOCK/CALENDAR W/ BATTERY BACK-UP
- * 100% IBM-AT COMPATIBLE BIOS
- * CABINET, 200 WATT P/S, & KEYBD..\$1595

OPTIONS TO COMPLETE YOUR "STATE OF THE ART" SYSTEM:

FLOPPY-HARD DISK 16 BIT CTRL ..	\$200
1.2MB FLOPPY DRIVE ..	\$139
26MB 65mSEC HARD DISK DRIVE ..	\$339
51MB 35mSEC HARD DISK DRIVE ..	\$809
85MB 28mSEC HARD DISK DRIVE ..	\$1398
140MB 30mSEC HARD DISK DRIVE ..	\$3319
20-60MB (2MB/MIN.) TAPE BACK-UP	\$795

AST 3G HERCULES MONO, COLOR
& ENHANCED COLOR GRAPHICS .. \$275

TATUNG 1380 13" ENHANCED DUAL
FREQ. RGB MONITOR .. \$479

TATUNG PRIVATE LABELS

IBM & MICROAGE MONITORS .. CALL

TATUNG 1422A DUAL FREQ.TILT
'N' SWIVEL AMB/GRN 14" 640x400 .. \$139

MACROTECH MSR-AT12MHZ,0-3MB

RAM, NO WAIT STATES .. \$185-\$625

NEC 1401 MULTISYNC .. \$515

PC-DOS 3.2 .. \$119

XENIX OPERATING SYSTEM .. \$395

LINK 125 GRN 14" TERMINAL , 1 YR WARR,

WY-50 "LOOK-ALIKE" .. \$385

ALLOY PC-SLAVE/168MHZ 8088,

1MB RAM PER USER .. \$595

ALLOY ATNX NETWORKING

SOFTWARE, UP TO 31 USERS .. \$189

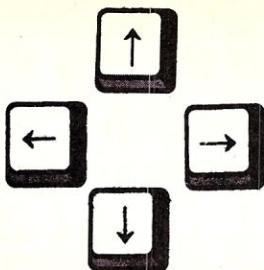
AUTO CAD ADE-0 .. \$259

800-528-3138 Orders Only

602-991-7870 Customer Support

TELEX 9103806778 SONE HUND





PC/Blue Report

by Hank Kee



The summer is nearly over, and the beach umbrellas and sailboats will soon be going back into storage. Many of us will be dusting off the personal computer to keep up with the latest commercial software releases, and of course the gems in the recent release of PC/Blue public domain and user support library volumes.

The trend in the recent submissions which I have been receiving are large scale systems normally requiring multi-diskettes, a color graphics adapter, and a fixed disk storage or equivalent. Many of these programs are quite good in functionality and complete in documentation. Without further ado, let's review the last batch of releases.

The STILL RIVER SHELL (Vol. 201) is a file and directory management utility. It stacks up to 20 of your last DOS commands for instant recall and can manipulate files and directories with single keystrokes. This file manager shows you realtime file, directory and disk information as you execute commands. It locates files easily anywhere on your system, graphically displays directory tree structures, allows you to view the contents of any file, sort files by name, extension, size or date, and backup large directories onto multiple diskettes.

The WAGNER FILE UTILITY (Vol. 201) manages large capacity disk drives that utilize the hierarchical file structure of MS-DOS version 2 and up. Function keys are used to execute the most used functions.

PC-WRITE (Vol. 202) version 2.6, by Bob Wallace, is the latest release of the most popular word processor in the library. Its features are comparable to many basic commercial WP's.

RELIANCE MAILING LIST (Vol. 203) is an advanced mail list system. The program is completely menu-driven with cursor keys to enter data and other such basic information. The Reliance Mailing List program maintains a data file of names and addresses and other information. Designed for a political campaign, it can be used by individuals, charitable organizations or small businesses as well. It is extremely easy to use; menus show all program functions. It prints two-across mailing labels and creates a *mailmerge* file to export data to other programs. You can select the names to print in a variety of ways and sort them by last name or zip code.

A Look At Some Of The New Public Domain Software Releases

PC-RIM (Vols. 204-206) is a full relational DBMS based on the popular public domain RIM-5 DBMS written several years ago. RIM-5 is the basis of a very popular commercial product. PC-RIM will run in a 512K PC and a hard disk is a practical necessity. Complete sources are presented so you can add to it, use its B-tree code (BT*.FOR), or whatever else you desire. It will run with or without an 8087, or 80287. This is basically a mainframe class DBMS.

Mr Bill and CK (vols. 207 - 210) are 2 diskette sets. Mr Bill is a billing system. CK is an income and expense tracking system. Each program consists of input preparation and report generation.

ANALYTICALC version X20.04D (Vols. 211-213) is the latest version of an integrated system. It includes all the basic functions of spreadsheet, database manager, and word processor. It is very complete in functionality. The registered version runs significantly faster. This is a state-of-the-art integrated system.

REC, Regular Expression Compiler (Vols. 214-215) is a very powerful symbolic manipulation language with some arithmetic capabilities; it is designed for the creation of compilers and operating systems. It has a very simple control structure consisting of four symbols: the two parentheses, the colon, and the semicolon. It is an attempt to reduce computer programming to its barest essentials. The resulting language is very concise, not unlike APL or TECO. There is a strong resemblance to TECO. The REC processor is remarkably compact, fitting into about 5Kbytes of memory in many machines (7.5Kbytes when floating point arithmetic operations are included), exclusive of working storage.

PIANOMAN (Vol. 216) is a program that allows you to play the keyboard of a PC

Editor's Note: Hank Kee is the librarian for the PC/Blue public domain software library. He is the person who collects, assembles, and checks all the software issued by PC/Blue and then compiles and edits them into the released volumes.

(or close compatible) like a piano. Since the keys are in two banks, it's more like an organ. The keys sound a tone as long as you hold them down, and stop when you release them. If you press the Record key, the tune you play gets saved in memory, and you can play it back at will.

Once you've recorded a tune, you can go into editing mode. In this mode you have a variety of editing commands, much like those of a word processor. You can insert and delete notes, adjust pitch and length, make global changes, and save/retrieve files. You can even mark blocks of notes to delete, copy, or move. Playing a tune with PIANOMAN is simple. You'll see a picture of the keyboard on your monitor, with the NOTES written on the keytops, instead of the normal letters. There are three octaves available at a time, and you can shift those octaves up or down using the up and down arrow keys.

PC-KEY-DRAW (Vol. 217) is a very advanced text and graphics generator. The supplied demo is a graphics show in itself. The litany of functions available are very impressive. This sophisticated program is a challenge to many commercial offerings.

The ever popular GENEALOGY on DISPLAY is now into release 4.0 (Vol. 218). This is a very popular program for keeping records of your family tree.

PC-HAM (Vol. 219) is a set of DATA BASE Amateur Radio programs including logging, dx'ing, simulation and model-

Copies of the PC/Blue printed software directory can be ordered from Micro/Systems Journal, Box 1192, Mountainside NJ 07092 (\$4 U.S., Canada & Mexico; \$6 foreign).

PC/Blue disks are available from the New York Amateur Computer Club, Inc., Box 106, Church Street Station, NY NY 10008. Price is \$7 per volume which includes media, postage, and handling. On foreign orders, please add \$2 per disk.

ing, and antenna positioning. The dBASE 2 programs perform database operations on logbook information.

TOXIC (Vol. 220) is a query system of a vast variety of chemicals indicating minimum toxic levels and remedies. A must for every nervous person living near industrious sites.

BLAQUE BOOK is a Database limited to Name, Address and Phone number. Its forte is in the printed output which is formatted in mini, alphabetized pages which you can assemble into a pocket sized address book. There is extra room under each letter for additional entries which you can later enter into your database. When you are ready for an updated address book, just print one.

The PC/Blue Library has now grown to 220 volumes. Recent releases include the following:

Volume 201

Still River Shell v1.21
file & directory management
by Bob Howard

Wagner File Utility v1.88
hierarchical file management
by Butch Wagner

Calc2
extended precision calculator
by Frank R. Webb

Volume 202

PC-Write version 2.6 word processor
by Bob Wallace/QuickSoft

Volume 203

Reliance Mailing List version 2.0
by Wm Meacham
GW Turbo - name and address database
by Gary Wooden

Volumes 204-6

PC-RIM database management system

Volumes 207-8

Mr BILL - billing system
by David M Alexander

Volumes 209-10

CK - income and expense tracking
by David M. Alexander

Volumes 211-13

AnalyticCalc version X20.04D
integrated system

Volumes 214-15

Regular Expression Compiler
by Gerardo Cisneros & H.V. McIntosh

Volume 216

PianoMan version 3.0
by Neil J. Rubenking

Volume 217

PC-Key-Draw version 3.0
Keyboard to Screen Graphics
by Edward H. Kidera IV

Volume 218

Genealogy ON Display version 4.0
by Melvin O Duke

Volume 219

PC-HAM Ham Radio Package
by Joe Kasser

Volume 220

TOXIC Gas Emergency Program
BlaqueBook N/A mini-database system
by Martin C Beattie

PolyMake

The Leading Make Utility

Are you still using a prehistoric Make? Now, step up to PolyMake, the most powerful and flexible Make utility available for programmers using MS-DOS. PolyMake is like an intelligent assistant that remembers how to rebuild a program when you change one part of the program. PolyMake will automatically invoke your compiler, assembler, linker, librarian or other tools to update a single program or entire software systems when you type — MAKE. PolyMake comes with built-in rules for rebuilding programs but you can also teach it new rules so you don't have to remember the file dependencies in your program. Advanced programmers prefer capabilities like fully recursive makefile processing and the ability to invoke PolyMake with special "flags" and macros that automate a whole range tasks. New Make users appreciate the Step-By-Step tutorial and intuitive commands. Handles source files written in any language. Requires DOS 2.0 & higher. Compatible with LANs, the IBM PC, XT, AT and other MS-DOS PCs. For complete details write for the POLYTRON Programmer's Catalog. **\$99**

PVCS

The Most Powerful & Flexible Source Code Revision & Version Control System.

The POLYTRON Version Control System (PVCS) allows programmers, project managers, librarians and system administrators to effectively control the proliferation of revisions and versions of source code in software systems and products. PVCS is a superb tool for programmers and programming teams. (A special LAN version is also available.) If you allow simultaneous changes to a module PVCS can merge the changes into a single new revision. If changes conflict, the user is notified. Powerful capabilities include: Stores and retrieves multiple revisions of text; Maintains a complete history of revisions to act as an "audit trail" to monitor the evolution of a software system; Maintains separate lines of development or "branching"; Provides for levels of security to assure system integrity; Uses an intelligent "difference detection" to minimize the amount of disk space required to store a new version. Requires DOS 2.0 or higher. Compatible with the IBM PC, XT, AT and other MS-DOS PCs. Single User version \$395. 5-station LAN version \$1,000, add \$500 for each additional 5 stations. **\$395**

TO ORDER: VISA/MC 1-800-547-4000, Dept. No. 329; Oregon/Outside US, 503-684-3000
Send Checks/P.O.s To: POLYTRON Corp. 1815 NW 169th Pl., #2110, Dept. No. 329, Beaverton, OR 97006

POLYTRON
High Quality Software Since 1982®

d/MULTI MULTIUSER dBASE for TurboDOS

TRUE File and Record Locking as easy as d-BASE-II. Unlimited users can perform the magic of dBASE in the program or interactive mode

- * TurboDOS 1.3 or 1.4
- * No Peeks or Pokes
- * System Date and Time Functions
- * Printspooler Controls up to 16 printers

Martian Technologies . . .
.CREATEing Multi-users from
Single-users around the world

CALL FOR DETAILS

Martian Technologies
8348 Center Dr., Ste.-F, La Mesa, CA 92041
(619) 464-2924

FREE
52 PAGE
CATALOG
AND APPLICATIONS GUIDE

CP/M
MSDOS

PDP — Plotter Driver Program
For multi and
single pen plotters **\$72.95**

XFER — Transfer Function Synthesis
Transfer functions/
Electronic circuits **\$72.95**

TEKCALC — Scientific Calculator
Program
Screen graphics/
Statistics/
Programmable **\$72.95**

COMCALC — Communications
Design Spreadsheet
Communications
budget calculator **\$72.95**

RIGHTWRITER — Report
Proofreader Program
Version 2 — Applies
2,200 rules of English **\$97.95**

AFFORDABLE ENGINEERING SOFTWARE

TRSOS
PCDOS

ACNAP — AC Network Analysis **\$72.95**

DNCAP — DC Network Analysis **\$72.95**

SPP — Signal Processing Program **\$72.95**

PLOTPRO — Scientific Graph Printing **\$72.95**

PCPLOT — High Resolution Graphics **\$72.95**

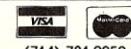
LOCIPRO — Root Locus Analysis **\$72.95**

ACTFIL — Active Filter Design **\$72.95**

STAP — Static Thermal Analysis **\$72.95**

MATRIC MAGIC — Matrix Manipulation **\$72.95**

BV Engineering
Professional Software



(714) 781-0252

2200 Business Way, Suite 207, Riverside, CA 92501 U.S.A.

The CompuMagic Utility Package

by Joseph A. Sabin Jr.

The CompuMagic Utility Package for CP/M-80, consists of 20 utilities and two installation programs. These utilities are a combination of unique user tools and simplifications of existing CP/M-80 commands/utilities. Each tool does what it is suppose to, and does it well. Following is a brief description of each tool in the package.

CMCOPY: This is an advanced copy program. It handles all the required functions of a copy utility as well as verifies a destination file overwrite, if requested. However, its best feature is undoubtedly its ability to use a source file for the file names you want copied.

COMPARE: I must say this is a disappointment. COMPARE checks two files for differences and tells you if they are different or the same. It works as promised but, if there were differences, I would like to see it report on the differences in content.

DS: Takes a document file and expands from single spaced to double spaced. It is designed to work with WordStar document files as well as standard ASCII.

ERASE: ERA for use from inside WordStar (or other programs that can call a program for execution). It works just like ERA with a few added features.

RENAME: REN for use from inside WordStar. It also is very similar to REN. These are nice to have and can prove a time saver.

SORT: Takes a text file and sorts the lines, placing the sorted text back into the same file or into a new file. The time for sorting 1000 names in a 12K file is 64 seconds; MicroPro's SuperSort needed 32 seconds. Timings were on an Altos 5-15 with dual 5 1/4" floppies with 4 MHz Z-80. The size of the file to be sorted is limited by the system memory. As I tested the sort facility with an oversize file, it would die with a controlled error message. When I reached a size close to the maximum size, it seemed to be working, but my patience gave out (it took too long).

WC: This program simply counts the number of words in a file. It is accurate and counted the words in the above sort file in 9.3 seconds.

FMINSTAL: This is the install program for all the above file handling programs. It is used to set the default flags for the programs.

MDIR & MDIRS: These programs are

very similar utilities and are paired together in CompuMagic's documentation. MDIRS returns the size of the file in Kbytes and the MDIR will list all directory entries in a sorted order which is under user control: by Extension, Size, User area.

DISKDIF: One disk has 65 files another has 63 files; which two are missing from the one? DISKDIF knows. This is most helpful for archiving or backing up a disk full of letters or other such electronic stuff that changes too often for the user to track. DISKDIF lists what is missing on the one disk, which can then be copied from the other.

DIRBAK: This is a directory listing of *.BAK saving 3 keystrokes.

DIRSPACE: How many directory entries are remaining on a disk? A client of mine kept crashing a program with a disk full error; yet there was always at least 100K of space remaining. This was due to the absence of any remaining directory entries. DIRSPACE would have informed her of that immediately. Indeed she had no idea how many files she could have on a disk.

UDIR: Ever misplace a file? If you have a hard disk and take advantage of the user areas, I am sure you have. UDIR will not only find it for you (i.e. which user area) but will tell you all the file names ambiguously by user area.

DINSTALL: All the directory tools list 24 lines by 80 columns before pausing. My TRS-80 has only 16 lines by 64 columns; DINSTALL allowed me to change the default size and all works well, no more wrapped lines of display and non stop scrolling off the screen.

A: When A.COM is placed on the B: drive and A;SC2 is typed instead of A:SC2, A.COM is run (CP/M thought that is what you meant). It then reads the command line and executes A:SC2 just as desired (NICE!).

CMAUTO: The program that creates programs to call other programs. Yes indeed, you can now create a program to directly begin a submit file from the command line. The created program is a .COM file and executes like any other except it calls the program specified when it was interactively created.

MINIERA: A small version of ERASE.COM for those strapped for disk space.

R/O: A brief command to replace

STAT FILENAME.EXT \$R/O. Instead you type R/O FILENAME.EXT. It will accept ambiguous file names.

R/W: The inverse of R/O.

SCREEN: This is great! SCREEN.COM sets up a file to grab all that is sent to the screen via BIOS or BDOS function calls. The screen is updated as normal and a copy is sent to the file specified.

TYPIT: This is a typewriter like program; it is also a disappointment. It buffers all input until a return is hit. This allows you to edit the line, but a typewriter's major advantage over a word processor is the ability to fill in forms. If the carriage does not move character by character you still cannot fill in a form except by trial and error. Change that and I think it will be a hit.

CONCLUSION

A well formed package of useful tools, some more so than others and some you may never use. The documentation covers all aspects of the tools to the extent of a reference manual. The beginning, however, is a tutorial. An important design feature of these tools is a unified set of command line parameters, one set of parameters for the programs. All things considered, this is a fine example of a compact tool kit.

CompuMagic will call you back if there is no one there to help when you call. In fact, I received a call back in fifteen minutes — after five O'Clock! In my conversation with the programs' author he told me the programs are written in hand optimized C, the very small programs, however, are straight 8080 assembler.

There are many disk formats available. CompuMagic uses Uniform (a disk formatting/copying utility with 96+ disk formats) to offer the maximum support for CP/M computers. Without question this is \$45.00 worth of software, and if you own a CP/M-80 machine, let's support those companies supporting us! [u]

Joseph A. Sabin Jr. is President of MicroFrame Associates, Inc. a company involved in semi-custom software design. He works primarily with financial & accounting data base software programmed in C, Pascal and assembler for CP/M-80 & 86 and MS-DOS.

REAL POWER PROGRAMS

**No fancy menus; No flashy graphics.
Just hard working programs that do what you want them to do.**

RECHARGE YOUR CP/M COMPUTER WITH THE CompuMagic UTILITY PACKAGE

CompuMagic's put 20 programs into one super CP/M Utility Package.

They're flexible.

They're fast.

They work.

Utility programs aren't sexy, but these programs have a way of making even dull tasks like file copying, renaming, sorting, and even erasing almost exciting. Directory listings, file and directory compare programs, turnkey program starters, screen capture and more. These programs are simple to learn, but powerful. The 20 programs are described in detail in the 50+ pages of documentation. Order the package now and start enjoying your CP/M computer again. Only **\$45.**

FIND IT FAST WITH SEARCH

CompuMagic's first utility for IBM PC users (for CP/M-80 & CP/M+ also) will find what you're looking for FAST! No need to index; search any text file at any time. SEARCH program or text files for one or more strings. See each match in context; you can see a window of lines or the word processor paragraph around a match. On-line help available at any time. Lots of easy-to-learn wildcards and boolean operators, even I/O redirection. Compatible with most popular word processors. Supports path names in MSDOS too. Documentation has lots of examples. Just **\$30.**

30-DAY MONEY-BACK GUARANTEE. All CompuMagic products come with a full, 30-day, money-back guarantee. If anything you buy doesn't work the way you want it to—and we can't make it right—we will refund your money.

NOT COPY-PROTECTED. CompuMagic products are never copy-protected.

ORDER NOW: Send Check or MO (Personal checks accepted) or call with your MC, Visa, or COD order. NO charge for UPS ground shipping. Add \$2 for UPS COD and \$5 for foreign orders. (MD Res. add 5% tax.) Be sure to specify operating system and disk format.

CompuMagic, Inc.TM

P.O. Box 437
Severn, MD 21144
(301) 969-8068

In the SIG/M Public Domain

by Stephen M. Leon

Steve Leon is the SIG/M Disk Editor. In other words, he is the person who assembles, compiles, and edits all of the SIG/M public domain software disks. Thus, he speaks with the greatest authority as to what is going on in the SIG/M public domain software area.

The car was in the shop waiting for new brakes, and so I wound up in the apartment building lobby about 6:30 AM waiting for a lift to the office. About five of us were sitting around talking and waiting, when the conversation turned to computers.

Seems one of my fellow tenants had just returned from a Taiwan business trip. While there he picked up copies of Wordstar 2000, Javelin, Lotus version 2 and Framework II. His total outlay for software (without copy protection) including manuals was less than \$50.00.

He described a bazaar-like building with about a hundred booths. What you wanted was made for you on the spot. You also could have custom programming or customized standard programs. For the most part, customers were local people.

What has Chinese software piracy got to do with SIG/M and public domain software? What they were doing in Taiwan apparently is legal under their copyright laws. Likewise, there are ads in all of the magazines for commercial operators who are selling or renting the SIG/M, and other public domain libraries. We could easily take either side of whether or not what these commercial seller/renters are doing is in violation of the copyright laws. A good deal of the material in the SIG/M library is copyright-protected and released only for non-commercial use. As a practical matter, however, what these seller/renters are doing serves a useful purpose in that it makes public domain software more readily available.

What really irks, however, are classified ads we see in some magazines from people selling one or more public domain programs as their own. It also irks to see people putting together collections of public domain programs from a variety of our volumes and selling them at \$25 per disk. Both practices violate author rights and the concept of making your programs available for public use public.

We tip our hats to the renters and the reasonable price sellers. They provide a service that benefits all. Watch out, however, for anyone who asks more than \$10 per disk or sells single programs that you know are available in the library. They both fit into the ripoff category.

In going through the overseas libraries to gather material for new SIG/M releases we note that they also include MS-DOS programs. Previously, we included only CP/M software in the SIG/M library. However, now that we have made it available in a variety of formats we are going to include some of MS-DOS programs in future releases.

What we will be releasing will be public domain programs with source code. Our primary interest will be in the utility category. The material we really like is the kind of short, practical programs that PC magazine publishes and we will see if we can get permission to release that code. We will release applications with source code too, such as the dBASE III version of the SIG/M Property Manager and Dr. Andy Bender's promised medical office package in KnowledgeMan.

There will not be any *begware* in this material. There also should not be any conflict with the PC/BLUE library. In fact, it seems to me that Hank Kee is so swamped with begware that the person who really wants to make a contribution to the public domain has no place to go. Therefore, if you have something to donate, even if it is PC/MS-DOS, consider the SIG/M library.

We finally released Irv Hoff's Improved Modem Program. It took three disks to hold IMP and all of its overlays, but we think it well worth it. We still prefer MEX, but IMP is a strong contender. Irv is also readily available for help on the CompuServe CP/M SIG.

One of the nice things about IMP is the broad range of overlay files available for it. The release includes overlays for just about every computer, including some that are hard to come by, such as the Apple 2c/e and Apple 3. Included in the release (Volumes 272 to 274) is KDM17, Irv's version of XMODEM.

BBS operators might be interested in trying KDM. I had the chance to use it when I called Irv's system to download a last minute update to the IMP overlay list.

The only suggestion we would make, is to rename it XMODEM. Everyone expects to have to key in XMODEM to get a file downloaded, but not KDM.

Irv also contributed a nifty little program (Volume 279) called XIZI. It translates Z80-to-8080 and 8080-to-Z80. He also updated FINDBAD, which we should soon have in the library.

The new releases also include three volumes of extracts from the Tesser Library of the Tesseract RCPM+, Dural, New South Wales, Australia (Volumes 275-277). There is some interesting material on these disks including a Z80 assembler and a program to read MS-DOS disks in CP/M and CP/M+. We have put out several of these programs. The problem is everyone has slightly different hardware. With a little luck, one of these MS-DOS readers will work on your system.

Another SYSOP program, RQ, (Volume 275) keeps a record of requests to put programs on the System. Also of interest, for those just getting a modem system working, is TINYTERM, a quick and dirty way of getting a modem operational.

Volume 276 contains a collection of ZCPR3 utilities from Australia. Volume 277 includes a backup program, a full screen editor and a telephone list program.

The featured program on Volume 278 is Z80DIS, a Z80 disassembler written in Turbo Pascal by Kenneth Gielow. Unfortunately, he has not released the source code, but we released it because of its many features. The program does an extensive analysis of the disassembled program's memory use and produces a break table based upon this memory usage analysis. It also gives you an address listing which itemizes all addresses referenced by the code, distinguishes types of label references, marks subroutines and jumps and produces a fully annotated cross-reference table. It is an interesting program well worth checking out.

Volume 279 is for those cataloging their disks. My favorites on the floppy, however, are LABEL, which prints a label for the disk, and TAIL, which shows the last few lines of a file.

Volume 280 contains some of Jon Dart's Turbo Pascal programs. You might find a use for his French and German vocabulary tests.

MAIL TO:

P.D.S.C. Co. (est. 1983) Suite L3, 33 Gold St. NYC 10038 212 732-2565
 West Coast Orders May Use Our Napa Phone 707 224-8931 4-10 p.m.

CP/M SOFTWARE Price List/Order Form

We recently acquired 80% of Osborne Computer Corp.'s Software stock, in a liquidation auction and can offer regular commercial grade software at prices far under usual wholesale. Most programs are copyable, and are in Osborne SD format, which we understand is readable by Commodore 128, Kaypro, Zorba and many others. Some of this software may be licensed for use on the Osborne Computer; you may want to obtain a second machine license or update from the publisher directly. Please ask about quantity/dealer pricing. We also have Osb manuals/parts. Circle items ordered.

Class A	Documate plus—indexes and builds table of contents. Deadline—solve murder game. ZORK-1—Adventure game.
\$17.00 each	Enumerator—adds line numbering to text files/Wordstar. Starcross—Galaxy game. Grammatick—checks grammar/syntax. Footnote—for text files/books. Math—adds calculator to W.S.
Class B	StartPac—tape tutorial intro to CP/M WS, Scalc, Basic. Forth v2.0—complete language for programming. MicroLink. DBase 11 tutorial set. Disk Doctor—repair kit for crashed files. Milestone—project mgr.. Mailman—for lists. Spellguard—corrects. Osborne—Diagnostic testor. Fortran. DD CP/M Utilities upgrade. BASCOM (No Manual).
\$28.00 each	
Class C	BSTAM—super modem, great for multi-file transfers. SuperSort—easy data manager, very capable. Personal Datebook. Wordpac: writer's packages for use with Wordstar, Orig. \$295.00 including Grammatick Spellguard, Math, Documate plus, Footnote. DataStar. Real Estate Mgr. SuperCalc. MBASIC.
\$39.00 each	
Class D	Osborne 1 5 Pac including Microsoft BASIC, SuperCalc, and Wordstar v2.24 with Manual—Complete text editor. \$95.00 ea. Personal Pearl—menu driven database manager, already set up. Any 3 of class C, or 4 of class B, or 8 of Class A.
Class E	DBase 11 w/Zip—the original relational database manager. \$175.00 ea. Executive set: Wordstar, Supercalc, C/MBASIC, CP/M 3.0, Personal Pearl. All titles are copyrighted by their respective publishers. OSB: Fieldservice Guide \$75.00. Exec. tech parts A&B \$75.00 (Excl. Bios) OSB: Gray Owner's Guide, Wordstar, Scalc, BASIC \$20.00. Exec. 4PAC Books \$50.00.

We also maintain a complete library of CP/Mug and SIG/M Public Domain CP/M Software. Presently about 90 meg/20,000 programs. Catalog \$4.00. 1 volume on 2 DD SS 0-1 Disks \$10.00. Post/Handling \$3.00 order.

We have Public Domain Libraries for TRS-2/CoCo, Commodore 64/Vic 20, Apple II e,c, plus, Macintosh, and I.B.M.-PC/Compatibles. Ask for Catalog. Please advise us of your needs. We will not be undersold. Terms: All sales are final due to closeouts.

Please Advise (Card #) _____ Amt. Enc. _____

Regards
P.D.S.C.

Name/Phone/Computer _____

Address/Zip _____

Volume 281 is from Phil Ridler in Zimbabwe (Rhodesia to you colonialists). Also on the disk is a Snoopy calendar that was donated at TCF. When you run it, don't panic. It grinds away for about 10 minutes writing a PRN file with no indication of what is going on. However, it does produce a multi picture calendar and can be updated.

We have been getting a lot of mail from M/SJ readers as well as requests for copies of the Commodore 128 sample disks. We have tried to answer all of them. One I would like to answer here is from David Danello of Blacksburg, Virginia. Dave comments on two of his favorite programs in the library: SWEEP by Dave Rand and Z80E by Rick Surwilo. Z80E is quite a handy debugger on Volume 239. SWEEP, on 152 (NEWSWEEP is on 209) is a file management utility that is a favorite of a lot of people. (I prefer VFILER because of its screen display.) The point Dave raises is that perhaps we should go further in publicly acknowledging fine contributions like these so as to encourage people to perfect or expand their software donations.

Dave Rand and Rick Surwilo and a lot of other people have made fine contributions to the public domain and we appreciate them all. We unfortunately can't send a personal thank you to every contributor. We do send contributors who request it two SIG/M volumes. We in SIG/M are volunteers just as the software

contributors are volunteers. Both groups are doing their best to promote computing. We wish we could personally thank all of our contributors.

SIG/M Volumes are available on 8" SSSD Disks (\$6 each, \$9 foreign) directly from SIG/M, Box 97, Iselin, NJ 08830. They are also available in many 5" formats (\$7 per volume; formats that require more than one disk add \$2 per volume). Printed catalogs are \$3.00 each (\$4.00 foreign). A disk catalog (Vol-0) is available at the standard disk volume price. The catalog volume is usually more current and more readily available than the printed catalog.

Many bulletin boards have the software for downloading and most new releases are available on the Compuserve CP/M SIG. Several commercial companies also sell, or rent, SIG/M software. For Apple CP/M users, the Philadelphia Area Computer Society, Box 477, Kulpsville, PA 19442, has most of the SIG/M volumes in Apple II format.

NEW SIG/M RELEASES**Volumes 272-274**

Improved Modem Program by Irvin M. Hoff

Volume 275

CROWECPPM Z80 assembler

RDMS231 Reads MS-DOS in CP-M & +

RQ14 SYSOP requests

TIMEZONE Time around the world

TINYTERM Minimal modem program

UNICON Conversion utility

Volume 276

LQ1500	Set Epson LQ-1500
ZCPR-WS	WordStar patch
ZCPR3CMD	ZCPR3 with ZCPR3CMD
ZDIR	ROM-resident commands Enhanced directory

Volume 277

BU12	Smart archive-backup
PED11C	Full screen text edit
PHONE3	Phone list utility
PROWRT12	Set C-Itoh 8510
SETATT	Set/clear attributes

Volume 278

FDUMP	File dump/patch utility
RPN	RPN calculator
Z80DIS21	Z80 Disassembler

Volume 279

COMP	File compare utility
FATCAT22	Disk catalog program
LABEL	Makes neat labels for disks
TAIL	Shows last few lines of a file
XIZI-2	Z80 ↔ 8080 translator

Volume 280

Turbo Pascal	Programming from Jon Dart
FRENCH	French vocabulary quiz
GERMAN	German vocabulary quiz
MACROSS5	Macro processor
MULTCOL	Single to multi column

Volume 281

Programs from Zimbabwe	
CMPCODE	compare two object code files
ZASMB	Z80 assembler
ZDIR	directory display
ZLOAD	HEX code loader
SNOOPY	Snoopy calendar

μ

Plu*Perfect Systems' DateStamper

by Gord Wiggins

I recently installed a Compu/Time S-100 clock board in my system and was anxious to find new applications for it. The ad for *DateStamper* in M/SJ caught my eye and sold me on having all my files date stamped and so I ordered it. *DateStamper* is available in many formats and comes with a neatly printed 65 page loose leaf manual. There is an up-date file on the disk as well. I found it hard to believe what I had received, not only the *DateStamper* program in two versions (8088 & Z80 code), but also several utilities I would have gladly paid \$50 for. You certainly get your money's worth with this package.

Plu*Perfect has made installation painless. No modifications of the BIOS required. *DateStamper* and its buffers are approximately 1K long and can be installed under CCP at the top of the TPA, or if you can find a spare K above your BIOS, you can install it there. The latter is the better choice as it leaves the TPA at full size and avoids any conflicts with bad mannered programs that hog the top of the TPA. Besides, I need all of my TPA for working on long programs with MAC and ED.

DateStamper can be run in two versions. One runs on almost any hardware at all and does not use a clock board (the *relative mode*). Instead of datestamping the files, it assigns a four digit number to each file. This number is incremented each time the file is accessed. So, by looking at the number, you can tell which one of several files, with the same name (on different disks), you worked on last.

If you have a clock board, you can use the *real time mode*. The result is the program, SSD, a *super directory* program similar to its public domain predecessor SD and ZCPR3'S XD and XDIR. SSD, however, displays the TIME and DATE on which each file was CREATED, ACESSED, or MODIFIED. It also shows file size and about a dozen other things as well as the CURRENT REAL TIME.

DateStamper and its utilities are *very* screen active. The only problem I had in getting it up was configuring my weird terminal to it. The terminal configure program presents a choice of 20+ terminals and, of course, mine was not there. Plu*Perfect offers to add yours if you send them the specifics. They also offer help by mail or phone, if you have trouble.

A Powerful Date Stamper For CP/M

Luckily for me, the terminal installation program is excellent allowing one to make up their own or modify any of the 20 or so already there. After about 5 tries, I got it to work with my terminal. The features you must have are: cursor addressing; clear-screen; clear-to-end-of-line; clear-to-end-of-frame; cursor up, down, left, right; and some form of highlighting (dim or inverse video is nice, but not essential). My dumb terminal does not have erase-to-end-of-line or erase-to-end-of-screen, so I had to leave out erase-to-end-of-line with the result that some garbage is left in certain parts of the display. Also, I substituted home-cursor and erase-screen for erase-to-end-of-frame. At least it works, and yes, I am saving my bucks for a proper terminal with all the neat features.

Next, I proceeded to get my clock board working with *DateStamper*. This turned out to be easy. The setup program presented a list of about 50 clocks; I had no idea there were so many. *DateStamper* is for all CP/M computers, not just S-100's. The list of clocks can be dumped to the printer and I was surprised to see more info come out on the printer than was on the screen. The screen display and the printer output of the clock lists were different. The 50 clocks were listed on the screen in two columns of 25, but lengthwise on the printer with additional info, both at the same time. Neat!

After selecting my clock board, the program asked for only two things, the CPU speed, and the clock board base address. After a warning not to test the clock unless you were sure of the addresses (might write to the disk controller instead) the clock was tested and there was the time and date with the seconds being updated in real time. The clock routine, now having been automatically written by the installa-

tion program, was then written to the disk under a name of your choice to be used later by the *DateStamper* installation program.

This was so easy I could hardly believe it. I had envisioned having to write assembly language clock driver routines. The manual does guide you on this, if need be.

Next, the installation program walked me through the modification of the *DateStamper* file for hardware specifics. It also allows one to put *DateStamper* above BIOS, if desired. Since my system is set up for ZCPR3 and I was not yet using the Resident I/O package, I had 1.5K where this was to go. So, I put *DateStamper* there. I also used ZCPR3's ability to run a multiple command line at cold boot time to install *DateStamper* at cold boot into RAM up above the BIOS.

DateStamper, when first installed at cold boot, or whenever, has the ability to run a file. You don't need SUBMIT on the disk as it is in *DateStamper*. Just tell *DateStamper* what command line to run after it is finished loading. However, your CCP must have the ability to run SUBMIT files. I had to put this back into ZCPR3 as I had taken it out to make more room and was running ALIAS instead (which is the same as SUBMIT, but is a disk-based ZCPR3 utility). I decided to have *DateStamper* give the command line LOG after loading.

LOG is a utility supplied with *DateStamper*. It keeps track of the time you spend on your computer doing one or more of nine activities which you select. Another program called TOTAL LOG gives a formatted print-out of your time in hours and tenths, month-by-month, for the entire year on each of the nine activities. I called mine Cbios mods, System mods, Word-processing, Data Base, Financial, Games, Basic, Forth, and Other. This is handy at income tax time to show how much you used the computer, and for what. This is an example of the great stuff Plu*Perfect supplies with *DateStamper*, at no extra cost.

You can LOG anytime to change activities. Before you shut down, you Log Off and your time is calculated and written to a file called DAILY.LOG which is used by TOTAL LOG to make the report. Simple and informative.

Another utility supplied with

Now! Automatic time and
date stamping for
CP/M® 2.2

DATESTAMPER™

Avoid
erasing the
wrong files!

Back up
files by time
and date!

"DateStamper...
is a real winner."

Bruce Morgen, Users
Guide, Jul-Aug. 1985

Write or call for
further information
(714) 659-4432

- Extends CP/M 2.2 to automatically record date and time a file is created, read and modified. DateStamper reads the exact time from a real-time clock if you have one, or records the order in which you use files each day.
- Date stamping information is contained in a separate file and read by our directory utility, SDD.COM. Powerful DATSWEET file management utility also included.
- Simple menu-driven installation. Large library of clocks supplied, with provision to add others. Many options configurable to your individual requirements.
- Disk prepared for date stamping are fully compatible with standard CP/M. DateStamper also runs with all versions of ZCPR, Z-RDOS and many other CP/M-80 modifications.

CP/M is a registered trademark of Digital Research, Inc.

Plu'Perfect Systems

8" SSSD, Kaypro, Osborne,	\$49
H/Z-89 formats	
(Most other formats, add \$5)	
Shipping & handling	\$3
California residents add 6% sales tax	
MasterCard &	
Visa accepted	

BOX 1494, IDYLLWILD, CA 92349

ENHANCING CP/M continued from page 33

```

; >inithist
; for z80 with simple bank select
; include listing.4a      ;global equates for history
;
; aseg      org      tpa
;
; move real program to memory above bank select ram
; ldi      hl,mstart
; ldi      de,nonbnk
; ldi      bc,pend-pstart
; ldir      jp      pstart
mstart:
; .phase nonbnk
; clear a page of ram to zeroes
pstart: ldi      hl,empty
        ldi      de,empty + 1
        ldi      bc,255
        ldi      (hl),0
        ldir
        ldi      a,bank2      ;select the second bank
        out     (bank),a
; set up loop counters for multi page clear
        ldi      de,hbuff
        ldi      b,high (hbuff)
clrloop: push    bc
; copy the clear page into the history buffer
        ldi      hl,empty
        ldi      bc,256
        ldir
        pop     bc
        djnz    clrloop
; set the current command pointer to point to
; the beginning of the (empty) history buffer
        ldi      hl,cbuff0
        ldi      (hbuff),hl
        ldi      a,bank1      ;select the first bank
        out     (bank),a
        ret      ;to CP/M's CCP without a warm boot
pend:
empty: ds      1
        .dephase
;
end

```

μ

AT LAST: Professional Typesetting Capability For PC Users

With **PC TeX™** — the best-selling full implementation of Professor Don Knuth's revolutionary typesetting program TeX.

FINEST Typeset Quality Printing From:

dot matrix laser phototypesetter

$$\sum_{i=1}^{\infty} \frac{1}{i} \begin{pmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix} \int_{-\infty}^{\infty} e^{-x^2} dx$$

WIDEST Range Of Output Device Drivers:

- Epson FX, LQ
- Toshiba
- Corona LP-300*
- Screen preview, with EGA or Hercules card
- HP LaserJet*
- Apple LaserWriter
- APS-5 phototypesetter

MOST COMPLETE Product Offering:

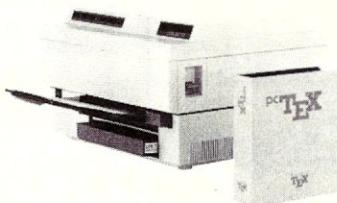
PC TeX (not copy protected) includes the following:

- Our specially written *PC TeX Manual*, which enables you to start using TeX right away.
- Custom "macro packages" that provide formats for letters, manuals, technical documents, etc.
- The *LATEX* document preparation system, a full-featured macro package for preparing articles, books, reports, etc., and *LATEX User's Manual*.
- AMS-Tex*, developed by the Amer. Math. Society for professional mathematical typesetting.

Site licenses, volume discounts, and interfaces to PC Paintbrush, PC Palette, FancyFont and Fontrix are also available.

PRICED FROM ONLY \$249.00!

(Printer drivers and interfaces additional.)



Laser printer,
fonts & software
from \$2995.00

For IBM PC/XT, AT or compatible, DOS 2.0 or higher, and 512K RAM. Hard disk required for printer drivers and fonts.
*HP LaserJet and Corona require additional interface boards.

For more information call or write:
Personal TeX, Inc.

20 Sunnyside, Suite H, Mill Valley, CA 94941 (415) 388-8853

This ad, with space for the photograph, produced by PC TeX. Typeset on the Epson FX80, the Corona LP-300 laser printer, and the Autologic APS-5 phototypesetter.

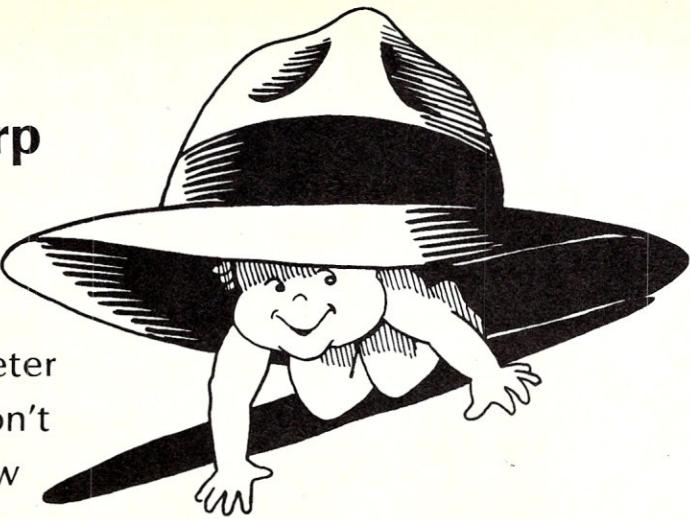
TeX is a trademark of the American Mathematical Society. Manufacturers' product names are trademarks of individual manufacturers.

C-terp

The C
Interpreter
You Won't
Outgrow

C-terp will grow with you as you progress from novice through professional to guru. Unbelievable, but true, the easiest-to-use C interpreter will provide you with the most advanced programming features for upward growth. Our exclusive **object module support** enables you to add libraries (like HALO, PANEL, Windows for C, etc., or your own homebrew libraries) to C-terp as you add them to your computing repertoire. Use C-terp as a microscope on your libraries! Flip a bit and allow our **software paging** (NEW) to handle those big jobs! There are no fixed-size tables to overflow, and C-terp can be configured for different screens and screen adapters (NEW). With multiple modules and full **K&R support**, we offer a dream C environment.

- Our new improved **configurable editor** competes with anything going.
- Speed -- Linking and semi-compilation are breathtakingly fast.
- Convenience -- Errors direct you back to the editor with the cursor set to the trouble spot.
- Symbolic Debugging -- Set breakpoints, single-step, and directly execute C expressions.
- Compatibility guaranteed -- batch file to link in your compiler's entire library. Supported compilers include: Computer Innovations C86, Lattice C, Microsoft C 3.0, Mark Williams C86, and Aztec C.
- Many more features including batch mode and 8087 support.



What Our Users/ Reviewers Are Saying

"...easy to use, powerful, and a timesaver."
"...we absolutely LOVE C-terp."
"...has restored my faith in interpreters."
"...a programmer's dream."
"...wonderful technical assistance."
"...increased our productivity by a factor of 40."
"...the best C product ever, in any category."

- Price: \$300.00 (Demo \$45.00)
MC, VISA

Prices include documentation and shipping within U.S. PA residents add 6% sales tax.
Specify compiler.

- C-terp runs on the IBM PC (or any BIOS compatible machine) under DOS 2.x and up with a suggested minimum of 256 Kb of memory. It can use all the memory available.

* C-terp is a trademark of Gimpel Software.

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261



Users' Group

Over 90 volumes of public domain "C" software including:

- compilers
- editors
- text formatters
- communications packages
- many UNIX-like tools

Write or call for more details

The C Users' Group

Post Office Box 97
McPherson, KS 67460
(316) 241-1065

ONE COMPUTER & TWO PRINTERS?

Stop Plugging & Unplugging!
use your choice of TWO printers from ONE
parallel cable with the simple flick of a switch.

\$65.95
Plus \$3.25
Shipping

30 Day
Money Back
Guarantee
Model #2100

One Parallel Plug In... Two Parallel Plugs Out!
Connect any Standard Centronics Parallel Cable directly to the switch. Two 3½" cables (included) attach to each printer.

Simply Call 1-800-TO-ASK-US! 8am-5pm PST
800-862-7587
(CA 415/567-4067)

To Order: Write or call Tipz Direct, Inc. Visa/MC or Money Orders please. Shipping within 48 hrs. via UPS.

Offering a FULL LINE of
Switches, Cables, & Accessories

P.O. Box 690,
San Francisco, CA 94101-0690



PLU*PERFECT DATESTAMPER

continued from page 84

DateStamper, and worth \$49.95 on its own, is DATASWEEP, a super combination of SWEEP, NSW and VFILER programs. It is all of these rolled into one with a very fancy screen display including a real real-time onscreen clock in the top right-hand corner ticking away. It would take 10 pages to describe the features of DATASWEEP in any detail and I am not going to attempt it. Just take my word for it, it is *fantastic*.

DateStamper has to have a file called !!!TIME&DAT as the *first* file in any directory on all disks. This is put on with another utility program called PUTDS.COM. This is easy to do, but takes a bit of time if you have to close 100 disks, of various sizes and densities, like I do. No real problem however.

Also supplied is a public domain utility called SAP51.COM and replaces earlier version of SORT AND PACK. So, don't use these earlier versions as they will mess up the DateStamper file. Good thing they supplied this as I was using SAP42.

My FATCAT has an option switch for disks with DateStamper on them. Remember, most catalog programs want a disk name starting with "-", usually the first file. Well, it's not any more. DateStamper is. So there may be some conflicts here with your catalog programs if they cannot handle the DateStamper file being first in the directory, or if they sort the directory and put DateStamper somewhere else, where it will not work.

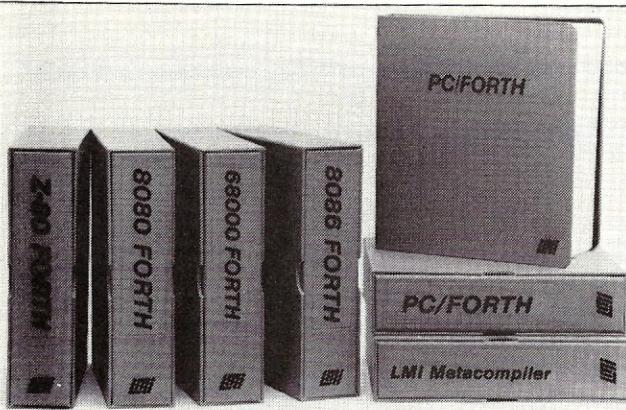
Plu*Perfect has modified a few public domain programs to work with DateStamper including MCAT and XCAT, MAKE, TOUCH and a few others and they sell these for the cost of the media plus shipping (\$12).

You can't go wrong buying DateStamper from Plu*Perfect Systems. The main program is excellent, does all they say it does and the utilities supplied with it are the icing on the cake. It is a treat to see fine software like this being produced for CP/M users instead of aiming at the mass MS-DOS IBM market. Congratulations Plu*Perfect, you have done us CP/M users proud.

**Not reading your
own issue?**

see page 90

TOTAL CONTROL with LMI FORTH™



**For Programming Professionals:
an expanding family of
compatible, high-performance,
Forth-83 Standard compilers
for microcomputers**

For Development:

Interactive Forth-83 Interpreter/Compilers

- 16-bit and 32-bit implementations
- Full screen editor and assembler
- Uses standard operating system files
- 400 page manual written in plain English
- Options include software floating point, arithmetic coprocessor support, symbolic debugger, native code compilers, and graphics support

For Applications: Forth-83 Metacompiler

- Unique table-driven multi-pass Forth compiler
- Compiles compact ROMable or disk-based applications
- Excellent error handling
- Produces headerless code, compiles from intermediate states, and performs conditional compilation
- Cross-compiles to 8080, Z-80, 8086, 68000, 6502, 8051, 8096, 1802, and 6303
- No license fee or royalty for compiled applications

For Speed: CForth Application Compiler

- Translates "high-level" Forth into in-line, optimized machine code
- Can generate ROMable code

Support Services for registered users:

- Technical Assistance Hotline
- Periodic newsletters and low-cost updates
- Bulletin Board System

**Call or write for detailed product information
and prices. Consulting and Educational Services
available by special arrangement.**

LMI Laboratory Microsystems Incorporated
Post Office Box 10430, Marina del Rey, CA 90295
Phone credit card orders to: (213) 306-7412

Overseas Distributors.

Germany: Forth-Systeme Angelika Flesch, Titisee-Neustadt, 7651-1665
UK: System Science Ltd., London, 01-248 0962
France: Micro-Sigma S.A.R.L., Paris, (1) 42.65.95.16
Japan: Southern Pacific Ltd., Yokohama, 045-314-9514
Australia: Wave-onic Associates, Wilson, W.A., (09) 451-2946

FORTRAN a structured language? Here's How!

New RF-77...a Ratfor-like preprocessor for PC FORTRAN

- WHILE, FOR and REPEAT loops.
- Free form input. No column 1, 6, & 7 to worry about.
- DEFINED constants and INCLUDED files.
- Outputs standard ASCII source code. May be compiled by any Fortran 77 compiler.
- Runs on any MS-DOS/PC-DOS equipped personal computer.

\$65.

Postage paid.
Texas residents add \$3.98 tax.

 Logical
Developments

P.O. Box 55798, Houston, Tx 77255

For VISA/MasterCard orders call:
1-800-835-2246, ext. 41

PC/FOCAL

FOCAL® language interpreter

This is the introductory programming language made famous on the PDP-8. All the familiar commands, Ask, Type, Set, For, If, Do, and so on, are present, plus a whole host of new graphics commands, DOS and ROM access commands, and disk I/O commands that tailor PC/FOCAL uniquely to harness the full power of your IBM PC/XT or compatible. Puts fun back into programming; you can do so much with so little code. Easy to learn and use, on-line help files, decimal and hex console I/O, direct memory color graphics, access to registers, access to all ROM BIOS and DOS facilities, joystick, bit manipulation, 8087 support built-in. Useful example programs on disk, including hex disk editor entirely in PC/FOCAL listable on one page. PC/FOCAL disk with user manual \$49.95 US postpaid. Requires PC or MS-DOS 2.1 or higher. UPS shipping \$5 (non-US shipping \$10)  

 Micro
Methods, Inc.

118 SW First St. - Box G
Warrenton, OR 97146
(503)861-1765

The UNIX File

by Ian F. Darwin

This column discusses the UNIX operating system. If you have comments or questions about UNIX or this column, please write to Ian Darwin at Box 603, Station F, Toronto, Ontario, Canada M4Y 2L8. If you have access to the uucp network, mail "ihnp4!darwin!ian". I can't always answer immediately, but I will get back to you; electronic mail gets answered first!

Every UNIX system comes with some electronic mail capability. Even if it's only *mail*, a program whose function has changed little since the early 1970's, the function is there to send mail to other people on the same UNIX system and, if the machine is on a network, to other people on the net. Several more ambitious mail programs have been written for UNIX over the years. Berkeley systems come with a mail system called *Mail* (called "cap-mail" since the initial "m" is capitalized, or Berkeley Mail for its origin). Berkeley Mail has been incorporated into System V Release 2 under the more reasonable name *mailx*. Rushing in where those with more wisdom had known not to tread, I once wrote my own mail program, only to abandon it when I discovered *MH*.

What's an *MH*? Unlike most mailers, it's not a large, all-in-one package. *MH* is a series of UNIX commands, invoked directly from UNIX shell that perform the common electronic mail functions. Some of the most common commands are *inc* to incorporate incoming mail into your *MH* directory, *show* to show (read) one or more messages, *comp* to compose a new message, *rmm* to remove a message, *repl* to reply, and *scan* to list the messages in an *MH* directory. A session with *MH* might appear as shown in the box.

Here I receive several messages with *inc*, look at one with *show*, reply to it with *repl* and *refile* it, look at another and delete it with *rmm*. *inc* lists the messages as it incorporates them, assigning each a number, and showing the date, sender, Subject: header, and as much of the first few lines of the message as will fit. *MH* provides convenient names such as first, last, cur to refer to particular messages instead of using the message numbers, and sensible defaults. The listing provided by *inc* can be recalled later with *scan*. *Show* can be configured to run its output

The MH Mail Handler

through a paginator such as *pg* or *more*. A really friendly feature is that *repl* and *comp* will invoke any editor you like to compose the text of the message; "prompter" is a simple editor-like program for entering short messages directly from the terminal. Since *MH*, like many UNIX subsystems, is controlled by a "profile" file, you can specify your choice of default editor, and many other defaults, separately for each program. Also, in any command that lets you edit a message, the mail headers (in-

cluding the recipient lists) are just text, and can be updated with your favorite editor. By contrast, Berkeley Mail makes you learn and use a series of special escapes to access header text.

And since you are always working with the UNIX shell, you can use the full power of shell syntax, such as

scan ! *lp*

to get a listing of all the mail in one directory sent to the line printer.

show all ! *grep* Fred

This is a (less-than-optimal) way of grepping all the mail in one directory for references to Fred. Since each message is a separate file in the UNIX file system, you can write shell files to manipulate your mail, such as an *edm* program to edit one or more messages.

Listing

```
% You have mail.  
% inc  
36+ 4/24 geoff@utcs.uucp <<Happy Birthday.  
37 4/24 sq!ian 2400b <<----- Forwarded  
38 4/25 msitxt!root <<Moving! Effective 5th May,  
39 4/26 root (/usr/adm/c search output <<adm owns  
40 4/26 ian (Ian F. Darw moving --> goodbye to uucp  
41 4/26 ian (Ian F. Darw hi; handshake failures;  
42 4/26 hoptoadignu (Joh Hoptoad is moving this  
43 4/26 ihnp4!uucp Warning From uucp <<We have  
% show  
(Message inbox:36)  
Return-Path: <utcs!geoff>  
From: geoff@utcs.uucp (Geoff Collyer)  
To: darwin!ian  
Date: 24 Apr 1986 0336-EST (Thursday)
```

Happy Birthday.

```
% repl -e prompter  
To: geoff@utcs.uucp (Geoff Collyer)  
cc: darwin!ian  
In-reply-to: Your message of 24 Apr 1986
```

Thanks!

```
What now? send  
% refile +f/geoff  
% show last  
(Message inbox:43)  
Return-Path: <ihnp4!uucp>  
From: ihnp4!uucp  
Subject: Warning From uucp  
Apparently-To: darwin!ian  
Date: 26 Apr 1986 1419-EST (Saturday)
```

We have been unable to contact machine

```
ncr-sd!mail ncrcae!sauron!xyzzy (Date 04/23)
```

The job will be deleted in several days
if the problem is not corrected.

Sincerely,
ihnp4!uucp

```
##### Data File: #####  
>From utzool!utcs!darwin!darwin.UUCP!ian Wed Ap<DEL>  
% rmm  
%
```

And, as important, MH is public domain. It written at the Rand Corporation, as an experiment in electronic mail. One of the authors of *Ethics and Etiquette for Electronic Mail* (reviewed a couple of UNIX files ago) was one of the original authors of MH. An early version, probably Version 4 of MH, was included with the 4.2 release of Berkeley UNIX. MH.6, the current version, is maintained at the University of California at Irvine, and a variant is scheduled to be included with 4.3BSD. Note that MH.6 is a large mail system, and includes support for several network types, mail-based "bulletin boards", and many other features. There's a lot of source code here, about 3 megabytes; it is not for the weak of heart, nor for the small of disk.

MH.6 is available free if you have access to the Internet (a collection of online networks including the older Arpanet). If not, you can often find a copy locally if you live in a major metropolis with lots of UNIX activity. If all else fails, you can get a copy by sending \$50 (U.S.) payable to the Regents of the University of California for the tape (including source and manuals) and a printed manual set. The UNIX manual pages are formatted with *troff*, and the rest of the documents (including an excellent introductory tutorial) are formatted with Knuth's TeX. Send your order to:

Support Group
Attn: MH Distribution
Dept. of Information & Computer Science
University of California
Irvine, California 92717
phone: 714-856-6852

Once you have electronic mail up and running, you can get listed in one or both of the "mailing lists" distributed by UCI; one for MH users and another for installers. Details come with the tape. If you're looking for the ultimate UNIX-based electronic mail package, get a copy of MH.

That's all for this month. I welcome electronic mail and letters on these and other topics, especially suggestions for future columns. Please write to me at Box 603, Station F, Toronto, Ontario, Canada M4Y 2L8. If you have UNIX mail access to the uucp network, mail "ihnp4!darwin!ian". I can't always answer immediately, but I will get back to you; electronic mail gets answered first! [P]

Ian Darwin is director of Research and Development for SoftQuad Inc, a company providing supported troff publishing software for UNIX. He and his wife live in a rural setting, north of Toronto.

It's 3 AM!



Do you know where your bugs are?

This C programmer is finding his bugs the hard way...one at a time. That's why it's taking so long. But there's an easier way. Use

PC-Lint 2.00

PC-Lint analyzes your C programs (one or many modules) and uncovers glitches, bugs, quirks, and inconsistencies. It will catch subtle errors before they catch you. By examining multiple modules, PC-Lint enjoys a perspective your compiler does not have.

- NEW: ANSI C extensions (enum, prototypes, void, defined, pragma), Microsoft keywords, and many additional checks.
- Full K&R C
- Use PC-Lint to find:
 - inconsistent declarations
 - argument/parameter mismatches
 - uninitialized variables
 - unaccessed variables
 - unreferenced variables
 - suspicious macros
 - indentation irregularities
 - function inconsistencies
 - unusual expressions
 - ... MUCH MUCH MORE
- User-modifiable library-description files for most major compilers.
- All warning and informational messages may be turned off individually.
- Indirect files automate testing.

NEW!!
Amiga - Lint
Special Introductory Price
\$98.00



PROGRAMMERS: Now your software can support over 150 video display terminals.

If you write software you need this book. It contains a tutorial on programming for video display terminals plus data sheets to support over 150 VDTs, including cursor positioning, clear screen, clear-to-end of line, row and column numbering, video attributes, function keys and much more! You'll find detailed information needed to support a wide range of VDTs immediately, getting your programs to market much earlier.

"Highly recommended for programmers."
— Jerry Pournelle, BYTE Magazine

335 pages, 6" x 9", softcover, ISBN 0-936158-01-8
15-day money-back guarantee. Texas residents add sales tax. Foreign residents specify and add shipping.
Send \$30 check, MC or VISA, to:

Atlantis Publishing Corporation
Dept. 205, POB 59467, Dallas, TX 75229

- Use it to check existing programs, novice programs, programs about to be exported or imported, as a preliminary to compilation, or prior to scaling up to a larger memory model.
- All one pass with an integrated pre-processor so it's very fast.
- Has numerous options and informational messages.
- PRICE: \$139.00 MC, VISA, COD (Includes shipping and handling within US) PA residents add 6% sales tax. Outside USA add \$15.00. Educational and quantity discounts available.
- Runs under MS-DOS 2.0 and up, with a minimum of 128Kb of memory. It will use all the memory available.
- Trademarks: PC-Lint(Gimpel Software), MS-DOS(Microsoft)

GIMPEL SOFTWARE

3207 Hogarth Lane • Collegeville, PA 19426
(215) 584-4261

FREE SOFTWARE FROM THE PUBLIC DOMAIN

User Group Software isn't copyrighted, so no fees to pay! 1000's of CP/M and IBM software programs in .COM and source code to copy yourself! Games, business, utilities! All FREE!

	Rent	Buy
IBMPC-SIG 1-390 DiskSides	410.	850.
IBMPC-BLUE 1-154 DiskSides	175.	435.
SIG/M UG 1-240 DiskSides	155.	650.
CP/M UG 1-92 DiskSides	45.	250.
PICO NET 1-34 DiskSides	25.	100.
KAYPRO UG 1-54 DiskSides	65.	200.
EPSON UG 1-52 DiskSides	65.	200.
COMMODORE CBM 1-28 DiskSides	25.	65.

Public Domain User Group Catalog Disk \$5 pp.
Send your check & specify computer format.
Rental is for 7 days after receipt, 3 days grace to return. Use credit card, no disk deposit.
Shipping, handling & insurance \$9.50 per library.
(619) 941-0925 Orders & Technical (9 to 5)
(619) 727-1015 24 Hr. 3 Min. Info Recording
Have your credit card ready!

National Public Domain Software

1533 Avohill Dr.
Vista, CA 92084
1-800-621-5640 wait
for tone dial 782542
AM EX.

Interrupt Borrowing With Turbo Pascal

see pages 34-40

Build an S-100 to PC Bus Converter

see pages 24-30

Implementing PC-DOS on Non-IBM Compatible Computers

see pages 28-30

TurboCharge Your 8086/8088 Computer

see pages 32-43

Also in this Issue
 Who Prints On The Printers
 Upgrading Computer I/O Boards
 Program Interfacing To MS-DOS
 Typesetting With Or Without A Typesetter
 The Compupro 10 Plus - A Review
 Complete Table of Contents on Page 3

35
42
54
61
45
54

Also in this Issue
 Interfacing to MS-DOS
 Loadable BIOS Drivers For CP/M
 Roll Your Own PC Clone
 C & Gforth On Disk-1 Controller
 Bringing Up ZCPR-3
 Complete Table of Contents on Page 3

32
65
36
46
42
51.00
25

Also in this Issue
 TurboCharge Your 8086/8088 Computer - Part II
 Assembling An AT Computer
 Build An S-100 Emulator
 Inter-system Processing
 Scientific & Technical Word Processors
 Complete Table of Contents on Page 3

32
44
69
66
50

Also in this Issue
 The Basic BASIC Forum
 Faster Floating Point Math
 Bringing Up EP/M-68K
 Programs Interfacing To MS-DOS
 Scientific Computer User
 Complete Table of Contents on Page 3

32-43

September / October 1985

Vol. 1 / No. 4

May / June 1985

Vol. 1 / No. 2

January / February 1986

Vol. 2 / No. 1

November / December 1985

Vol. 1 / No. 5

The only magazine by and for advanced micro users.

Micro/Systems Journal is the independent magazine for advanced software and hardware hackers who use MS/DOS, CP/M, MP/M, TurboDOS, Concurrent-DOS and Xenix. An important resource for programmers who use C, Turbo Pascal, Forth, Lisp and Assembler. It also covers hardware based on the S-100, PC and AT busses, as well as powerful single board and multi-user systems. Its name — MICRO/SYSTEMS JOURNAL!

Each issue is packed with practical articles...in depth tutorials, reviews, hints...the latest information on SIG/M, PC/Blue, C-User Group and PC-SIG Public Domain Software. Advanced topics such as 16-bit and 32-bit microprocessors, high performance data acquisition systems, interfacing to peripherals, patching application software packages for improved performance...and lots more! Strictly technical info...no fluff...stuff to keep every hacker up-to-date on the ever-changing micro technology.

Edited by Sol Libes, former editor of Microsystems, former columnist in Byte, PC Tech Journal, PC Week, Computers & Electronics and other magazines and author of 15 books. An authority in the field since the early seventies.

Here are examples of topics covered in recent issues.

- Build an S-100 to PC-Bus Interface
- Loadable Drivers for CP/M
- Structured Programming with Microsoft's M80 Assembler
- Review of Three 16-Bit Lisp & Prolog Packages
- Local Variables in Forth
- Installing ZCPR3
- Build a PC Clone for Under \$1,000
- Interfacing to MS-DOS
- Assembly Language Extensions For MS-Basic
- Reviews of Scientific Word Processors, Concurrent DOS and Turbo Pascal Version 3.0

**Save Up To 27% Off Newsstand Price
TAKE ADVANTAGE OF SPECIAL LIMITED TIME OFFER
so hurry and mail your subscription now.**

YES! Start my subscription!

[] 1 YEAR*	\$20.00
[] 2 YEARS**	35.00
[] 1 YEAR (Canada & Mexico)*	28.00
[] 2 YEARS (Canada & Mexico)**	52.00
[] 1 YEAR (other foreign)*	35.00
[] 2 YEARS (other foreign)**	64.00

* 6 issues **12 issues

Checks must be payable in U.S. funds by a U.S. bank

Name: _____

Address: _____

City: _____ State: _____ Zip: _____

make check payable to: MICRO/SYSTEMS JOURNAL
Box 1192
Mountainside, NJ 07049

THE SOFTWARE DIRECTORY

When contacting software publishers please mention that you read about them in Micro/Systems Journal.

Program Name: SMK (Seidl Make Utility)**Requirements:** MS-DOS System

Description: An automatic software product generation utility to update and track every module in a program. When changes are made to any module, SMK will recompile or reassemble changed modules and modules effected by the changes, and relink the modules in the proper order. Works with any compiler, assembler and linker. Supports in-line and blocked comments. Includes a formatter to make changes to source code modules automatic. Features include: proprietary dependency analysis algorithm to analyze all dependencies before rebuilding any files; understands complicated dependencies involving nested include files and source and object code libraries; high-level dependency definition language supports parameterized macros, local variables, constants, include files, command line parameters, line and block comments; batch source code editor allowing automatic source file updating; full pathname and directory support.

Price: \$140

Publisher: Seidl Computer Engineering, 1163 E. Ogden Ave., Suite 705-171, Naperville IL 60540, (312)983-5477

Program Name: RPG II**Requirements:** MS-DOS System

Description: RPG language compiler implementation based on the IBM S/34 RPG II. Provides sequential, direct and indexed file (ISAM) processing, linkage to user-coded assembly language routines and disk sort/merge utility.

Price: \$1500

Publisher: Software West Inc., 637 So. Wrightwood St., Orange CA 92669, (714)542-5202

Program Name: ModemMail**Requirements:** CP/M2.2 system

Description: A full-service electronic bulletin board and automated mail network system with terminal and communications functions. User can reply to, copy, or print mail messages sending messages via the lowest cost path. Enables computer to act as a gateway between timesharing or a commercial electronic mail service and a ModemMail network or other timesharing service.

It can encrypt mail messages and restrict user access to areas containing sensitive or confidential information. Includes a customization program for easy installation and configuring of the system.

Price: \$249

Publisher: Autosoft Inc., 166 Santa Clara Ave, Oakland CA 94610, (415)658-2881

Program Name: LATTICE TOPVIEW TOOLBASKET**Requirements:** PC or compatible

Description: For software developers writing IBM Topview applications. Consists of a library of more than 70 C functions to control window, cursor, and pointer operations. Also includes printer control, cut and paste and debugging functions, and sample source programs.

Price: \$250; Source code add \$250.

Publisher: Lattice Inc., Box 3072, Glen Ellyn IL 60138; (312)858-7950; TWX 910-291-2190.

Program Name: PC-Lint**Requirements:** Any PC/MS-DOS System

Description: A C language diagnostic facility. Analyzes C programs, across multiple modules and reports bugs, glitches and inconsistencies. Resembles "Lint" running under UNIX but has more features and is turned to the 8086 environment. Includes full K&R support, one-pass, very fast operation, use of the large memory model internally with all tables expandable, configurable to arbitrary architectures and special Lint-style comments to suppress errors. Is delivered with user-modifiable standard library descriptions for most major compilers.

Price: \$139

Publisher: Gimpel Software, 3207 Hogarth Lane, Collegeville PA 19426.

Program Name: LOCIPRO**Requirements:** PC/MS-DOS, CP/M-80 or TRSDOS system

Description: Provides control system and electronic engineers a simple means to quickly determine closed loop system stability from open loop transfer functions. Solves the locus of roots for systems up to 26th order and ten loop elements. Output data can be vectored to a line printer or data files. All program inputs are free format and menu driven. Out files are compatible with other BVE products adding transient analysis and high resolution graphics.

Price: \$72.95

Publisher: BV Engineering, 2200 Business Way #207, Riverside CA 92501

Program Name: TUTORI/O**Requirements:** Any CP/M2.2 System with 24K memory minimum

Description: Invokes any file functions via a single keystroke. Useful for assembly-language program developers who want to see the File Control Block and I/O buffer displayed and altered after each function. Customized to run on licensed system only but not copy protected. 30-day full money-back guarantee.

Price: \$31.

Publisher: Logic Associates, 1433 W. Thome, Chicago, IL 60660, (312)274-0531

Program Name: MPS

Requirements: MS-DOS or CP/M system. MS-DOS system must have Microsoft macro assembler and linker

Description: A user-extensible language via user defined PROC's. Eliminates need for redundant coding. File I/O includes full B-Tree ISAM, Random, and Sequential with unique features. Has simplified screen formatting and data editing, fast BCD arithmetic (up to 36 digits), program chaining with common data, access to operating system functions, powerful string manipulation. Allows in-line assembly code and generates object code.

Price: MS-DOS version, \$149.95; CP/M version, \$249.95 (includes macro assembler and linker). Manual, \$39.95 credits toward purchase.

Publisher: System Facilities Inc., Box 7079, Charlottesville VA 22906, (804)977-5245, or Box 5664, Lacey WA 98503, (206)456-0628.

Program Name: DSD86 & DSD87**Requirements:** PC or compatible

Description: Full screen symbolic debugger. Features window with user-controlled screen layout (6 different display types including instructions, registers, stack, memory and source). Keyboard can be customized. Recursive macro facility allows extensions to commands.

More than one window can be opened and positioned at any line or address in program. Scroll up and down, trace, single step, and set breakpoints.

DSD87 adds support for 8087 math coprocessor showing internal 8087 registers and flags.

Price: DSD86 - \$69.95; DSD-87 - \$99.95

Publisher: Soft Advances, Box 49473, Austin TX 78765, (512)478-4763.

C CODE FOR THE PC*source code, of course*

Concurrent C	\$45
LEX	\$25
YACC & PREP	\$25
Small-C compiler for 8086/88 .	\$20
tiny-c interpreter & shell . . .	\$20
Xlisp 1.4 & tiny-Prolog . . .	\$20
C Tools	\$15

The Austin Code Works

11100 Leafwood Lane

Austin, Texas 78750-3409

(512)258-0785

New Products

When contacting vendors please mention that you read about them in Micro/Systems Journal.

S-100 HARDWARE

ENCLOSURE FOR LARGE MULTI-USER SYSTEMS

The 5820-S is a heavy duty S-100 tower style systems enclosure. It includes a ferroresonant-based power supply rated at 50 amps (+8V), a highly efficient cooling design, 20 slot motherboard with active termination (adjustable) and independently fused connectors. The enclosure (38" high x 12.1" wide x 26.25" deep) has a 3 position key switch and LED bus and drive indicators. There are 32 DB-25 openings on the rear panel.

The enclosure can house and power a wide assortment of drives with power sequenced to be applied to the drives 3 seconds after power is applied to the bus. Price is \$1,495. From Para Dynamics, 7895 E. Acoma Dr, Scottsdale AZ 85260, (602)991-1600.

PC COMPATIBLE HARDWARE

Z-80 COPROCESSOR SUPPORTING UP TO 16 USERS

The Turboslave-PC (tmk) is an 8Mhz Z80 coprocessor card for PC/XT/AT type systems. It contains 128K RAM (with parity), 2 serial ports, a port-mapped FIFO for Master-Slave communications, on-board diagnostics, monitor EPROM and the SLR Z-80 Assembler software.

The board can be supported by multi-user TurboDOS(tmk) or configured

as a coprocessor under MS-DOS. Up to 16 Turboslave-PC cards can be supported under TurboDOS, each running as a dumb terminal, with full record locking and security. The Turboslave-PC can be used as a background processor.

The price is \$495, from Earth Computers, Box 8067, Fountain Valley CA 92728, (714)964-5784.

DEBUGGING SYSTEMS

Periscope I, Version 2.00, is a debugging system for software developers who need full crash recovery capability, including protected debugger software. It includes a break-out switch and 16K write-protected RAM board. An installation program loads crucial debugger software into the board, then write-protects it so that runaway software programs cannot corrupt it. The break-out switch allows recovery of hung systems and locked keyboards or just to interrupt programs to see what is going on. It can be used with compilers that generate line number references in their object files (e.g. Computer Innovations, IBM, Lattice, and Microsoft C and IBM and Microsoft Pascal, to see symbols, line numbers and actual object code. Traceback enables user to view previously-executed instructions. An in-line symbolic assembler is included as well as support for EGA cards. Users can view text files, change windows while debugging, break I/O port access, and customize Periscope with their own programs via user exits.

Periscope-I includes the memory board, debugger software, break-out switch, 150-page manual and quick reference card. Comes with a 30-day money-back and costs \$295, from Data Base Decisions, 14 Bonnie Lane, Atlanta GA

30328, (800)722-7006 or (404)256-3860. A demo disk (\$5) is available.

For those not needing the memory protection feature there is Periscope-II for only \$145. It includes the break-out switch and all the other feature of Periscope-I.

OTHER PRODUCTS

68020 DEVELOPMENT SYSTEM

Gimix Inc., 1337 W 37th Pl, Chicago IL 60609, (312) 927-5510, has announced the GMX-68020, a multi-user, multi-tasking, demand-paged, virtual memory system based on the Motorola 68020, 32-bit microprocessor. It includes 512K RAM, 3 serial ports (19.2K max), a 19M hard disk and 1Mb floppy. It uses the UniFLEX VM Operating System V. Written in assembler, it claims to be several times faster than standard Unix systems and C-source compatible. Motorola's 68020-Bug is included. Languages available includes: Ada, C, Cobol, Fortran, Basic, Assembler, Franz Lisp, and Prolog.

IN-CIRCUIT EMULATOR

A full speed in-circuit emulator called "I.C.E. BOX" for Z-80, 8085 and NSC800 systems. Consists of a palm-size unit, cable and plug assembly that plugs into the system's processor socket; then plug a terminal or other computer into the RS-232 port on the I.C.E. Box. A single command tests system RAM, ROM and exercises all I/O ports, memory addressing and analyzes code for runtime bottlenecks. Up to 65,535 hardware breakpoints may be set. Can trace program execution, skipping subroutines if needed, and can assemble and disassemble code. Optional software is available to automate test and repair operations. Price is \$600 plus shipping (\$5 U.S., \$25 overseas). From Softaid Inc., Box 2412, Columbia MD 21045, (301)792-8096.

Coming in the next issue!

*Speeding Up The PC/XT

*Assembling A PC/XT Clone - Part II

*Intercepting XIOS Calls

*Concurrent DOS - Part II

*Recovering PC-DOS Files

*Reviews of the PC's Limited 12Mhz

AT Clone, MatchCad & The Z-System

Announcing a New Product...

OPEN SPOOL PRINTER RIBBON THAT RE-INKS ITSELF

FOR USERS OF

**Okidata and Other
Open Spool Printers
Including Teletype**

MONEY BACK GUARANTEE



OUR SELF-INKING DEVICE WILL LAST 15 TIMES
THE RIBBON YOU ARE NOW USING.

- 15 TIMES LESS RIBBON CHANGES
- NO MORE MESSY HANDS

CONTROLLED PRINTOUT DEVICES, INC.

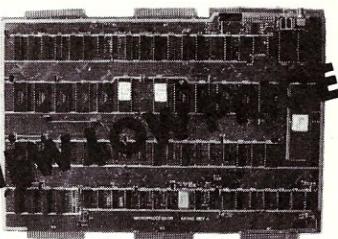
P.O. BOX 869, BALDWIN ROAD

ARDEN, NC 28704

(704) 684-9044 • TELEX: (FILMON-AREN) 577454

CONTACT US BY MAIL, PHONE OR TELEX AND WE WILL FORWARD YOU A BROCHURE

M68000 SINGLE BOARD COMPUTER



On board 6-10 MHz CPU, 20K RAM, 32K EPROM,
two RS-232, 16-bit port, 5-counter/timers
expandable via Memory/FDC Board.

M68K CPU (bare board)	\$ 89.95
M68K CPU A&T (6MHz)	\$495.00
MD512K Memory/FDC (bare board)	\$ 89.95
MD512K Memory/FDC (128K)	\$495.00
FDC/Hard Disk interface option	\$150.00
M68KE Enclosure w/power supply	\$249.00
M68K Monitor EPROM's	\$ 95.00
M68K Macro Cross Assembler	\$195.00
4XFORTH OS w/assembler, editor	\$295.00
CP/M 68K OS w/“C” compiler	\$395.00

EMSEducational
Microcomputer
SystemsP.O. Box 16115
Irvine, CA 92713
(714) 854-8545

PC SPEEDUP

NEW PRODUCT! — PC-Sprint — \$89.95

Exec-PC is the specialist in low cost PC speed enhancement.

PC-Sprint™ - Works with any PC, color or mono. 4.77, 6.67, 7.38mhz. Speed and reset switch.
Plug-in installation

Surprise!™ - from Maynard Electronics. Run any 8088 PC at 9.54mhz. Change speeds from Keyboard. No expansion slot used. 100% compatible.

V20 8mhz - Replaces 8088 up to 8mhz (6 or more, \$12.76 or \$10.00 with PC-Sprint)

V30 8mhz - Replaces 8086 up to 8mhz (6 or more, \$12.76)

V20/V30 Tech manual - 224 pages, sold with chip order only

Dealer Inquiries Invited.

All products include chip removal tool, user guide, 90 day warranty, free BBS trial subscription, no extra charge for credit card or shipping. (\$10.00 minimum on charge orders.)

Exec-PC, Inc. P.O. BOX 11268
SHOREWOOD, WI 53211
ORDER HOTLINE RECORDING INFO ORDER BY MODEM
(414) 242-2173 (414) 242-2165 (414) 964-5160



Pascal Tutorial

Introducing a terrific new tutorial you can use to learn to program in the modern programming language Pascal. Pascal uses structured methods for program definition leading to an easy to write, and easy to understand and maintain program.

- If you are not a Pascal programmer, this tutorial will clearly describe the most fundamental aspects of Pascal programming. It assumes you know nothing about programming and leads you step by step from the fundamentals to the advanced features of Pascal.
- If you are already somewhat experienced in Pascal programming, this tutorial will lead you into using those aspects of Pascal that you have been ignoring. A clear definition of records, pointers, dynamic allocation, and even linked lists are given along with examples of each.
- There are 71 Pascal source files on a disk that you compile and run after examining them using the explanations given in the preprinted instruction manual.
- This tutorial is especially written for TURBO Pascal 2.0 or 3.0 but could be used with any Pascal compiler since the majority of the programs are written in standard Pascal as defined by Niklaus Wirth.
- The last chapter includes several example programs to illustrate program development methods. These programs are also on the disk and can be compiled and run as further examples of programming in Pascal.
- Requires MS-DOS or PC-DOS 2.0 or later and can be run on any IBM-PC or compatible computer. Equipment requirements are only as needed for your compiler.
- Not Copy-Protected and guaranteed to be effective. If you are not satisfied, we will gladly refund your money if requested within 30 days.
- Sent postpaid anywhere in the USA. Send check or money order for \$24.50 to:

Coronado Enterprises
12501 Coronado Ave NE
Albuquerque, New Mexico 87122

Phone (505) 293-5464 Evenings and Weekends

Micro/Systems Journal™ Back Issues

The following back issues are available at \$4.50 per copy (\$6 foreign, cash, Postal Order, or U.S. bank check) including shipping. If ordering 3-9 copies deduct 10%, 10 or more copies deduct 15%. Orders can be combined with back

issues of Microsystems for discount. Send to M/SJ Box 1192, Mountainside, NJ 07092.

MARCH/APRIL 1985 (Vol. 1, No. 1): Bringing up CP/M Plus, Assembly Language Extensions for MS-Basic, New Tricks for CP/M2.2, Building an IBM/PC or XT Clone, Extended Single Density Storage, Variable Size Arrays in C; REVIEWS: dBase-III and 16-Bit Lisp & ProLog-Part I.

MAY/JUNE 1985 (Vol. 1, No. 2): Build an S-100 to PC-Bus Converter, Interfacing to MS-DOS Part-1, Loadable Drivers for CP/M2.2, Roll Your Own PC-Clone, Bringing up ZCPR-3, C & Godbout Disk-1 Controller, Writing Translation Programs in C and Turbo Pascal; REVIEWS: 16-Bit Lisp & Prolog-Part II.

JULY/AUGUST 1985 (Vol. 1, No. 3): Structured Programming With Microsoft M80 Assembler, Local Variables in Forth, Interfacing to MS-DOS Part-II, Data Translation with Turbo Pascal, Implementing Sets with Bit Operations in C, A Unix Mail List System; REVIEWS: Scientific & Technical Word Processors-Part I, Macrotech MI-286 S-100 CPU Card, Slicer System, Concurrent PC-DOS, Coherent Operating System.

SEPTEMBER/OCTOBER 1985 (Vol. 1, No. 4): Who Prints on Printer with CCP/M, Interrupt Borrowing with Turbo Pascal, Upgrading CompuPro I/O Boards, Program Interfacing to MS-DOS Part-III, SIG/M & PC/Blue Public Domain Software Distribution Points, TurboDOS INSTALL program; REVIEWS: CompuPro 10+, TEX.

NOVEMBER/DECEMBER 1985 (Vol. 1, No. 5): TurboCharge Your 8086/8088 Computer, Faster Floating Point Math with C, Bringing up CP/M-86K, Program Interfacing To MS-DOS Part-IV, Advanced Machine Level Interface Techniques For Turbo Pascal, Context-Independent Macros for C, Build Your Own PC Program Library Using Public Domain Software, dBase-II Speed Techniques Part I, Tuning Unix Program; REVIEWS: Peak 68K8-CP, Eureka CP/M-80, Scientific Workstations.

JANUARY/FEBRUARY 1986 (Vol. 2, NO. 1): Implement PC-DOS on Non-IBM Compatible Computers, Part-I; TurboCharge Your 8086/8088 Computer, Part-II; Assembling An AT Clone; Build An S-100 EPROM Emulator, Transferring Files Between CP/M and MS-DOS Systems; Program Interfacing to MS-DOS Part V; dBase-II Speed Techniques; Unix Public Domain Software; REVIEWS: Scientific Word Processors-Part II; Turbo-DOS PC; Turbo Pascal Support & Enhancement.

MARCH/APRIL 1986 (Vol. 2, No. 2): Implementing PC-DOS On Non-IBM Compatible System Part-II (Conclusion), Program Interfacing To MS-DOS Part-VI, Rolling Your Own PC/XT/AT Clone, Converting From CP/M To MS-DOS On The PC, A C Subroutine For Parsing Command-Line Arguments, Arithmetic Aberrations With dBase-III and the dBase Developer's Release, The Marriage of TEX and Postscript; REVIEWS: C Interpreters, C Source-Level Debuggers, Lomas Data Products' S-100 PC, CompuPro S-100 PC Video Board.

MAY/JUNE 1986 (Vol 2., No. 3): Build An S-100 HD64180 CPU Card, Configuring MS-DOS, Writing MS-DOS Device Drivers (Part 2), Building An AT Clone, Adding A RAM Disk To A CP/M System, Obfuscated C Code Contest, Sorting With Turbo Pascal, More dBase Speed Techniques, Using an RCPM System; REVIEWS: Concurrent DOS, MEX-PC, ConIX, FirsTime.

JULY/AUGUST 1986 (Vol 2., No. 4): Power to the PC, Creating A Copy Protected Program, Build Smart Keyboard Interface, Writing An MS-DOS Memory Disk Device Driver, Declare & Define C Variables in One File, More Loadable BIOS Drivers For CP/M; REVIEWS: Phoenix' PFIX-Plus Debugger, Holliston Challenger XT-186, Magnum Digital PRO-180 & FD-100, XPIP, PMATE, Ryan-McFarland Fortran, WATCOM, Watfor-77, MicroGlyp SciPlot, Softech Matrix Calculator, Personal TEX.

Microsystems Back Issues

Some issues of the old Microsystems magazine are still available in very limited quantities. They are \$4 per copy (\$5.50 foreign, cash, Postal Order, or U.S. bank check) including shipping. If ordering 3-9 copies deduct 10%, 10 or

more deduct 15%. Orders can be combined with M/SJ back issues for discount. Send to: MS/J, Box 1192, Mountainside, NJ 07092.

1984

MARCH: MS-DOS Overview Part-I, Enhancing MP/M-II Part-2: adding login, date and time functions, Julian date Conversions, I/O changes for Fortran-80, CP/M Mass Renaming by Filetype, CP/M BIOS public domain enhancements, Power Failure Backup for S-100 systems; REVIEWS: MS-DOS for CompuPro From Computer House, Lomas Lighting One, Dual Systems SIO4, MagicBind, SAL/80K.

JANUARY: Enhancing MP/M-Part 1; Installing MP/M; Add Concurrency to MP/M; Two Users on CP/M; Relocating Assemblers & Linkage Editors-Part 3; S-100 Wait States; REVIEWS: MP/M-8/16, ProComp-8, Paragraphs Game Board, ProLog.

1983

NOVEMBER: Intro to 80286, 68000, and 16032 Microprocessors; Intro to Local Area Networks-Part 2; Extended Memory Management for older S-100 Systems; Notes on Microsoft Fortran-80; Building S-100 Parallel Ports; REVIEWS: CompuPro CPU-68K, System 8/16, Xenith Z-100, Nevada & Ellis Computing Fortran.

OCTOBER: Intro to Local Area Networks, Part-1; Build Low-Cost LAN; Build S-100 Bubble Memory Card; Use Radio Shack Model 100 portable with a CP/M system; Write Menu Driven Utility for Setting

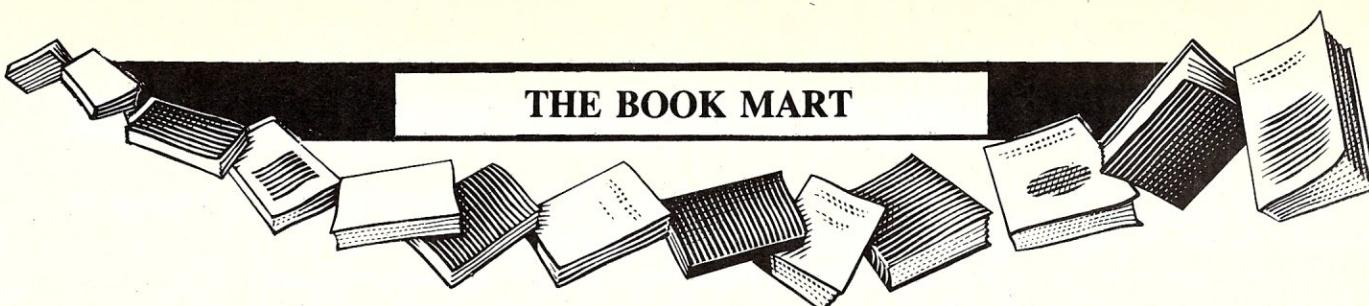
Printer Options; North Star Improvement; True Z-80 Random Number Function; Hide Code in Basic REM statement; Machine Code loader for MBasic; Increase Single-Density Disk Formatting; Relocating Assembler & Linkage Editors, Part-2; Run MX-80 with North Star; User Group Directory; CP/M-86 Versus CP/M-80; REVIEWS: CP/NET, QBAX, S-Basic.

AUGUST: XERA Program; Logging-On CP/M; WordStar Date/Time Patch; Find Location of Variable in North Star Basic; Prevent System Crashes During Warm Boot; Enhance Spreadsheet Print Files; Plotting Package-Part 3; Run WordStar under TP/M; 50-line Text Formatter; Using the LU Utility; User Areas under CP/M; REVIEWS: Stiff Upper Lisp, MuLisp-80, Supersoft Lisp, Cromenco C-10, Access Manager, Fancy Font, Computime SBC-880 S-100 card.

JULY: Using RCPMs; RCPM Directory; PIP Data Between Computers; Toward Smarter Modem Programs; Interface MX-80 via Parallel Interface; Digital Audio On CP/M System; Customize CP/M CBIOS; Plotting Package Part-2; REVIEWS: DRI PL/I-86 and PL/I-80, S-100 PMMI MM-VTI.

JUNE: Plotting Package Part 1; Drive HP Plotter; Laboratory Graphics Applications; Console Keypressed interrupts; Customize Continue on next page

THE BOOK MART



The following books can be ordered through Micro/Systems Journal. Prices include shipping (NJ residents add sales tax) with prices shown for U.S., Canada and Mexico. Add \$5

for other countries. Send checks for full amount payable in U.S. funds by a U.S. bank. We

cannot bill or take credit cards. Allow 15-20 days for order to be processed. Sends orders to Micro/Systems Journal, Box 1192, Mountainside NJ 07092.

NEW WORTHWHILE BOOKS

80386/80286 Assembly Language Programming by W.H.Murray & C.H.Pappas. Also covers 80387 and 80287 math coprocessors. Includes many examples and techniques. \$20.95

Micro To Mainframe Links by R.F.Kopeck. Covers file server, LAN, Gateway, and Bridge technologies. \$19.95

Turbo Pascal Programmer's Library by K. Jamsa & S Nameroff. A collection of more than 250 routines and complete programs as well as advanced programming techniques. \$19.95

analysis of MS-DOS, PC-system architecture, ROM-BIOS service routines and much more. \$20.95; Companion disk \$65.95

Assembly Language For IBM PC DOS by Joe Dorner. Book \$20.95; Disk \$40.95

PC/BLUE Software Directory

Catalog listing contents of public domain software disks in PC/BLUE library. Includes cross-reference index by application and other useful information for PC/MS-DOS public domain software users. \$5

CP/M

Programmer's Guide To CP/M Edited by Sol Libes. Collection of all CP/M articles from Microsystems magazine (Jan 1980 to February 1982). \$14.95

Following books, by Dave Cortesi, are recommended for learning more about CP/M.

Inside CP/M \$27.50

Inside CP/M-86 \$18.95

The Osborne/McGraw-Hill CP/M User's Guide Includes Concurrent-DOS, Concurrent CP/M-86, DR/Net, DR-EDIX, and DR/Talk. How to use these packages more effectively and trouble shoot hardware and software problems. \$19.95

16-Bit SIG/M Public Domain Software Directory An index, by application, of all 16-bit CP/M-86, MP/M-86 & CP/M-68K in the SIG/M library (almost 200 programs), cross-indexed to volume number so you can locate them easily. \$2

OTHER BOOKS OF INTEREST

Designer's Guide to Disk Drives by Teja & Gonnella. \$ 20.95

Discover Forth by T. Hogan \$15.95

Operating Systems Design: The XINU Approach by Comer. How to write a Unix-like operating system with much of source code included. \$34.95 (hard cover)

S-100 BOOK BACK IN STOCK

We have located a small number of copies of **Interfacing To S-100/IEEE-696 Microcomputers** by Sol Libes & Mark Garetz. This is the definitive book on the subject. It is no longer being published and when this quantity is exhausted there will be no more. So hurry and send in your check. \$24.95

PROGRAMMER'S REFERENCE CARDS

The following are concise, comprehensive, and very complete reference cards designed with the programmer in mind. They are printed in two colors on 8½" x 11" laminated plastic and 3-hole punched suitable for insertion into loose-leaf binders. \$5.95 + \$1 each (U.S., Canada & Mexico; Foreign \$3) Shipping & Handling; ordering 3 or more cards \$.50 S&H (\$1.50 each Foreign).

C Language Card

UNIX Shell Card

IBM/GW Basic Card

WordStar Card

8086/8088 Card

Z80 CPU Card

POCKET REFERENCE CARDS

The following are concise comprehensive reference cards folded to fit into a standard shirt pocket. They are printed in two colors on heavy weight card stock.

CP/M Programmer's Reference Card.

Covers all CCP commands, utility commands, system programming and interfacing information as well as other CP/M information. \$2, includes shipping & handling (U.S. Canada & Mexico; \$3 Foreign).

WordStar Reference Card. Covers all

page, cursor, printer, edit and format control commands, logically organized. \$1, includes shipping & handling (Foreign \$2).

1982

NOVEMBER/DECEMBER: CP/M vs. MS/DOS; CP/M-86 vs. MS-DOS; Intro to ADA Part 2; Virtual Disk for NorthStar; CP/M Program Auto-execute; Macros & Macro-Assemblers; REVIEWS: Janus, Aztec-C, C/80, Morrow S-100 M26 Hard Disk System, Teleram S-100 Bubble Memory Card, Jade S-100 Bus Probe.

1981

NOVEMBER/DECEMBER: Introduction to the C Language, Virtual Segment Procedures, Little-Ada Part-II, A Disk Alignment Routine, Northstar DOS/BIOS Directory & File Conversion using UCSD Pascal; REVIEWS BDS-C, Small-C, Tiny-C and Whitesmiths-C; Tarbell Double-Density Disk Controller.

TURBO PASCAL Language

Using Turbo Pascal by Steve Wood. For new Turbo Pascal (V3.0) programmer. Contains lots of code examples. \$20.95

Programming With Turbo Pascal by David Carroll. By the author of the Turbo Pascal Corner in this magazine. Includes disk with limited version of Turbo Pascal and lots of sample programs; 310 pgs, \$34.95

IBM-PC/XT & COMPATIBLES

Inside IBM PC: Access to Advanced Features & Programming by Peter Norton. Detailed

Microsystems Back Issues

Continued from previous page

Wordprocessor Keyboard; WordStar Patch for H-19/Z-19 Terminal; Relocateable Code; REVIEWS: Graftalk, Jes S-100 Graphics Controller, ZCPR2.

APRIL: IEEE-488 Tutorial; Interfacing to Lab Instruments; CP/M-86 System in Lab; Implementing CP/M + PART II; Build Simple S-100 Card Extractor; Macros & MacroAssemblers; REVIEWS: Pickles & Trout S-100 488 Controller; CP/M Utilities; Morrow Decision I.

CLASSIFIEDS

Micro/Systems Journal accepts Classified Ads. The charge is \$6/line (3 lines minimum); 40 characters max./line. Three times frequency

\$15/line; six times \$25; non-profit clubs \$2/line. Logos, special type, etc. are extra charge. Check must accompany ad copy. Send to Micro-

/Systems Journal, Box 1192, Mountainside NJ 07092.

\$995 LAP-TOP CP/M MACHINE

NEC 8401A with serial and parallel interfaces, 300b modem built-in, battery powered, and 16 line x 80 col. display. **Interface Technology, Inc.**, Box 3040, Laurel MD 20708, (301)490-3608.

FOR SALE: S-100 BOARDS

Z80 CPU, 64K RAM, 16K RAM, floppy disk controller, SASI interface. All boards with manuals. Unbeatable prices. Call: 213-541-1058, evenings.

JONTEL BUSINESS SYSTEM

The Jontel Business System is a five journal accounting package based on a dual disk CP/M computer. Included with the seven disk package are mailing and inventory programs with 100+ pages of documentation on disk. 8" & 5" disk formats available. \$69.95. Moonlite Computer, 707 Edge Hill Rd., New Bern, N.C. 28560. Source code available for custom installations, compiled CB80. (919)638-6976.

FOR SALE: CompuPro 20 slot S-100 enclosure/motherboard, Paradydynamics model 2200D dual 8" enclosure, 2 Mitsubishi model M2894-63 8" drives, Advanced Digital 4MHz Z-80 SBC, US Robotics S-100 1200B modem, ADDS model VWPNT terminal, Comrex CR-1 Daiseywheel printer, CP/M 2.2 and SCP/80 front end, Pascal MT+ and Aztec C compilers, 100 8" disks, 2 8" flip'n'files. Running system, used < 200 hrs. Package = \$2000. W, 804-253-4006; H, 804-693-5897.

CompuPro System FOR SALE: Near-new home system, CPU-Z, RAM16, DISK1, INTERFACER 4, CCS 12 slot mainframe, 2 Mit M2896-63 half-ht 8" drives, Tel 510+ term, software + CP/M. Present new value: \$2780. Sell for \$1795 + UPS. 703-463-6793.

CP/M Software. Choose from 400 Public Domain volumes. 100 page catalog \$7.50. New CP/M 2.2 Digital Research Manual \$19.95 + \$3.00 S&H. Send SASE for Flyer. ELLIAM ASSOCIATES, 6101 Kentland Avenue, Suit M, Woodland Hills, Ca 91367

2/4/2x

FOR SALE: CompuPro CPU-Z 6MHz \$125, Disk 1A and CP/M-80 \$475, 64K static RAM \$97. ALL equipment is in new condition. James Walker, 1205 North Main, O'Fallon MO 63366.

S-100 USERS!!!

S-100 BOARD BANK buys, trades, sells, consigns ancient, modern S-100 cards, etc. **SELLERS:** send full description, BEST price. **BUYERS:** savings on S-100 cards from only \$49; send \$4.37 or more to subscribe to our whimsical, official irregular organ and ad extraordinaire, **THE STATEMENT**. Mailorder only. Write **S-100 BOARD BANK**, Box 344 Mailorder Dept. M2, Olympia WA 98507. 2/1/6x

DISK CONVERSION SERVICE

Over 300 CP/M and MS/DOS (PC-DOS) formats. Originals and copies returned in 48hrs. Reasonable rates. Personal service. For information and list of available formats, write: RH Associates, Box 7035, Silver Spring MD 20907, or call (301)587-6230.

Advertiser Index

Allen, Emerson & Franklin	75
Anderson Techno-Products	24
AndraTech	43
Applied Innovations	42
Atlantis Publishing	89
Atlatz International	12
Austin Code Works	91
BD Software	41
BG Computer Applications	9
BV Engineering	79
Boston Software Works	42
Competitive Edge	25 - 27
Compu-Magic	81
Computer House	42
CompuPro/Viasyn	1
Computime	24
Controlled Printout Devices	93
Coronado Enterprises	93
C Users' Group	86
Custom Software Systems	50
C Ware Corporation	55
D & W Digital	56
Data Base Decisions	57
DataDesk International	C3
Digital Decisions	61
Digital Research Computers	68
Earth Computer	40
Echelon	73

Educational MicroComputer Systems	93
Electronic Control Technology	71
Exec-PC	93
Fulcrum Computer Products	19
Gimpel Software	86 & 89
Hawkeye Graphics	43
Integrand	9
Intelligent Computer Design	58
Intercontinental Micro	35, 37, 39
Laboratory Microsystems	87
Logical Developments	87
Lomas Data Products	13
Macrotech International	C2
Martian Technologies	79
M Global	59
Microcomputer Systems Consultants	50
MicroDynamics	52
Micro Methods	52 & 87
Micromint	65
Micro Smith	70
Micro Supply Organization	48 & 49
Micro/Systems Journal Renewal	90
Micro/Systems Journal Books	95
Micro/Systems Journal Back Issues	94
Mitek	60
Mix Software	2
Mycroft Labs	42
National Public Domain Software	89
Night Owl	60
Peak Electronics	70
Performics	45
Personal TEX	85
Plu*Perfect	85
Polytron	79
Poor Person Software	70
Port-A-Soft	18
Public Domain Software	83
QCAD Systems	69
Rapid Systems	11
Rational Systems	74
R & L MicroServices	8
S-100	76 & 77
Semidisk Systems	C4
Servo Computer	43
Slicer Computers	44
SLR	8
SofCap	15
Sunny Hill Software	18
System Facilities	23
Tee Kay Software	44
Thinker's Apprentice	5
Tipz	86
TSF	44
Viasyn/CompuPro	1
Western Wares	44

RISK FREE
30-DAY MONEY-BACK GUARANTEE

NOW FOR YOUR IBM PC, XT, AT, AT&T, PC Jr. AND COMPATIBLES *

"THE PC HARDWARE BARGAIN OF THE YEAR"

Chuck Humble — Computer Editor



OUR NEW PC 8700 AT-STYLE KEYBOARD AND
BORLAND'S SUPERKEY® SOFTWARE

THE PC 8700 AT-STYLE KEYBOARD INCORPORATES EVERY IMPROVEMENT
IBM DESIGNED INTO THEIR ADVANCED "AT" KEYBOARD. QUALITY SO
GOOD, IT'S BACKED BY AN UNPRECEDENTED TWO-YEAR WARRANTY!

Advanced design features include:

- Separate Numeric Keypad simplifies data entry
- Status Lights for Cap, Num and Scroll Lock helps eliminate typing errors
- Extra-wide Control and Return Keys for ease of use
- Return Key placed in "standard typewriter" position for touch typists
- Selectric layout for Office Standardization
- Positive Tactile Feedback
- Keystrokes rated for 50 million keystrokes
- Made in USA
- 2 Year Warranty

BORLAND'S AWARD-WINNING SUPERKEY INCLUDED! GET THIS BEST-SELLING KEYBOARD ENHANCER SOFTWARE — VALUED AT \$69.95 — AS PART OF THIS AMAZING OFFER AND CAPTURE THE POWER OF MACROS!

SuperKey's macro capability is remarkable. It literally turns a thousand keystrokes into one.

- Exclusive screen data Cut & Paste
- Screen Privacy and Protection
- Time & Data Stamping
- ProKey Compatibility
- Data Encryption
- Keyboard Lock
- Keyboard layout customization
- Integration with SideKick
- Pull-Down Menus
- On-Line Help and much, much more!

ORDER NOW — CALL TOLL FREE 800-826-5398; IN CA 800-592-9602

EASE RUSH ME _____ KEYBOARD/SUPERKEY BUNDLES AT
19.95 EACH. PLUS \$10 SHIPPING AND HANDLING IN THE U.S.A.*

ENCLOSED IS MY CHECK FOR \$109.99 FOR EACH BUNDLE
(\$116.45 FOR CA RES. — SALES TAX INCL.)

VISA MC NUMBER _____

P. DATE _____ SIGNATURE _____

ME _____

DRESS _____

TY _____ STATE _____ ZIP _____

TELEPHONE _____ COMPUTER TYPE & MODEL _____

* PC Jr., AT&T and Corona computers require adaptors available at a nominal fee. Call for pricing.

Both
Hardware & Software

\$129^{.95}

Introductory
Offer #99.95

INCLUDES BORLAND'S SUPERKEY
AWARDED "BEST OF 1985" BY
PC MAGAZINE



RETAILS ALONE FOR \$69.95

CRITICS' CHOICE

"Who Can Pass Up a Deal Dept?"

John C. Dvorak, Infoworld, 3-10-86

"The best part of the keyboard is the feel. It is ideal!"

PC Productivity Digest, 5-86

"The keyboard's construction is solid . . . also quick and responsive to the touch. It is ideal for anyone who wants to or has to replace their PC keyboard . . . At \$129.95 DataDesk's keyboard is a bargain. The PC 8700 keyboard with SuperKey at that price is a steal."

Computer Shop Talk, 3-3-86

"DataDesk International has designed a sturdy and handsome keyboard that has a tactile response . . . and bundled with SuperKey is the PC hardware bargain of the year."

Charles Humble—Computer Editor, The Oregonian

"DataDesk's is genuinely a nice keyboard . . . It feels nice . . . I can type much faster and much more accurately on it than most others . . . and it's inexpensive. It costs only \$129, compared to around \$300 for the Keytronics 5151."

Harry Newton, Publisher, Teleconnect, 5-86

TELECONNECT
TEST DRIVE
SCORECARD

The DataDesk
PC8700 Keyboard

TECHNICAL PERFORMANCE	10
EASE OF USE	10
DOCUMENTATION	10
OVERALL RATING	10

NEUTR
RIVE
CARD

The KeyTronics
5151 Keyboard

TECHNICAL PERFORMANCE	9
EASE OF USE	9
DOCUMENTATION	9
OVERALL RATING	9

TELECONNECT'S
CHOICE

datadesk
INTERNATIONAL

7650 Haskell Ave.
Van Nuys, CA 91406
818-780-1673

The Peak of Performance



SCALE THE HEIGHTS OF PRODUCTIVITY

Sure, you've proven that in your hands a computer is a productive tool. But if you haven't teamed up with a SemiDisk you have heights yet to climb!

IT'S NO MERE RAMDISK

SemiDisk has been leading the way for Disk Emulators since their inception. If you've seen RAMdisks you know what it's like to load programs in an

instant, and read or write files without delay. Unlike alternatives, the SemiDisk offers up to 8 megabytes of instant-access storage while leaving your computer's main memory free for what it does best - computing!

KEEP A GRIP ON DATA

Go ahead, turn off your computer. Take a vacation. With the battery backup option, your valuable data will be there in the morning even if you aren't. You'll sleep better knowing not even a 5 hour blackout will sabotage your files.

NEW LOWER SEMIDISK PRICES THAT WON'T SNOW YOU UNDER

	512K	2Mbyte
IBM PC, XT, AT	\$495	\$995
Epson QX-10	\$595	\$995
S-100, SemiDisk II	\$799	\$1295
S-100, SemiDisk I	\$595	—
TRS-80 II, 12, 16	\$695	\$1295
Battery		
Backup Unit	\$130	\$130

Software drivers available for CP/M 80, MS-DOS, ZDOS, TurboDOS, and VALDOCS 2.

SEMDISK

SemiDisk Systems, Inc.
P.O. Box GG, Beaverton, Oregon 97075
503-626-3104

Call 503-646-5510 for CBBS/NW, and 503-649-8327 for CBBS/Aloha, all SemiDisk equipped computer bulletin boards, (300/1200/2400 baud) SemiDisk, SemiSpool trademarks of SemiDisk Systems.